

**APLIKASI PEMBELAJARAN INTERAKTIF
UNTUK MATA PELAJARAN
TEKNOLOGI INFORMASI DAN KOMUNIKASI
PADA SMP NEGERI 3 PALLANGGA**



SKRIPSI

Diajukan untuk memenuhi salah satu syarat guna mencapai gelar
Sarjana Komputer pada Jurusan Teknik Informatika
Fakultas Sains dan Teknologi
UIN Alauddin Makassar

Oleh :

SUTRISNO

NIM : 60200107109

UNIVERSITAS ISLAM NEGERI
ALAUDDIN
MAKASSAR

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI (UIN) ALAUDDIN
MAKASSAR
2012**

PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan di bawah ini:

Nama : Sutrisno

NIM : 60200107109

Jurusan : Teknik Informatika

Judul Skripsi : Aplikasi Pembelajaran Interaktif Untuk Mata Pelajaran
Teknologi Informasi Dan Komunikasi Pada SMP Negeri 3
Pallangga

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar merupakan hasil karya saya sendiri dan bukan merupakan pengambil alihan tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran sendiri.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut sesuai ketentuan yang berlaku.



Makassar, 14 Agustus 2012

Penulis,

Sutrisno
NIM : 60200107109

PERSETUJUAN PEMBIMBING

Pembimbing penulisan skripsi saudara **Sutrisno**, dengan NIM : **60200107109**, mahasiswa Jurusan Teknik Informatika pada Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar, setelah dengan seksama meneliti dan mengoreksi skripsi yang bersangkutan dengan judul, ***“Aplikasi Pembelajaran Interaktif Untuk Mata Pelajaran Teknologi Informasi Dan Komunikasi Pada SMP Negeri 3 Pallangga”***, memandang bahwa skripsi tersebut telah memenuhi syarat-syarat ilmiah dan dapat disetujui untuk diajukan ke sidang *Munaqasyah*.

Demikian persetujuan ini diberikan untuk proses selanjutnya.



Makassar, 14 Agustus 2012

Pembimbing I

Pembimbing II

Mustikasari, S. Kom., M. Kom

Abdul Wahid, ST., M.Kom

PENGESAHAN SKRIPSI

Skripsi yang berjudul “**Aplikasi Pembelajaran Interaktif Untuk Mata Pelajaran Teknologi Informasi Dan Komunikasi Pada SMP Negeri 3 Pallangga**” yang disusun oleh Saudara Sutrisno dengan NIM : 60200107109, Mahasiswa Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Alauddin Makassar, telah diuji dan dipertahankan dalam sidang Munaqasyah yang diselenggarakan pada hari Selasa, 28 Agustus 2012 M dinyatakan telah dapat diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer dalam Jurusan Teknik Informatika dengan beberapa perbaikan.

Makassar, 28 Agustus 2012 M
10 Syawal 1433 H

DEWAN PENGUJI

- 
1. Ketua : Dr. Muhammad Halifah Mustami, M.Pd (.....)
 2. Sekretaris : Wasilah, ST., MT (.....)
 3. Munaqisy I : Ridwan A. Kambau, S.T., M.Kom (.....)
 4. Munaqisy II : Faisal, S.T., M.T (.....)
 5. Munaqisy III : Drs. Hamzah Hasan M.Hi (.....)
 6. Pembimbing I : Mustikasari, S.Kom., M.Kom (.....)
 7. Pembimbing II : Abdul Wahid, ST., M.Kom (.....)

Diketahui oleh :

**Dekan Fakultas Sains dan Teknologi
UIN Alauddin Makassar**

Dr. Muhammad Khalifah Mustami, M.Pd
NIP. 19710412 200003 1 001

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Assalamu'alaikum Wr.Wb

Puji syukur penulis panjatkan kehadiran Allah swt., atas limpahan rahmat dan hidayah-Nya serta kesehatan yang tak terhingga, sehingga skripsi dengan judul “Aplikasi Pembelajaran Interaktif Untuk Mata Pelajaran Teknologi Informasi dan Komunikasi Tingkat Sekolah Menengah Pertama” dapat terselesaikan. Shalawat dan taslim tak lupa pula penulis haturkan kepada junjungan Nabi Muhammad saw. yang telah membawa kita semua dari alam kegelapan menuju alam terang benderang, dan dari dunia kebodohan menuju dunia yang penuh dengan ilmu pengetahuan.

Skripsi ini diajukan untuk memenuhi salah satu syarat dalam meraih gelar Sarjana Komputer pada Jurusan Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri (UIN) Alauddin Makassar.

Penulis menyadari bahwa penyusunan skripsi ini tidak akan terselesaikan tepat waktu tanpa adanya bantuan dari berbagai pihak. Oleh karena itu, dengan segala kerendahan hati penulis mengucapkan terima kasih yang sebesar-besarnya atas segala bantuan berupa dukungan, arahan, saran serta bantuan moril dan materil kepada penulis. Untuk itu penulis menghaturkan banyak terima kasih kepada :

1. Teristimewah kepada keluarga penulis, kedua orang tua, Muhtar M, S.Pd dan St. Norma, SE yang telah merawat, membesarkan, membimbing, mendidik, dan mendo'akan penulis selama menempuh pendidikan. Serta Kedua saudariku, Nurhasanah dan Nurlia atas segala doa dan bantuan baik moril dan materil kepada penulis. Semoga Allah swt. senantiasa melimpahkan kesehatan dan kesejahteraan bagi mereka, Amin.
2. Bapak Prof. Dr. H. A. Qadir Gassing, H. T, MS. selaku Rektor Universitas Islam Negeri (UIN) Alauddin Makassar.
3. Bapak Dr. Muhammad Khalifah Mustami, M.Pd selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Alauddin Makassar.
4. Bapak Faisal, S.Kom., M.Kom selaku Ketua Jurusan Teknik Informatika serta Bapak Nur Afif, ST., M.T selaku Sekretaris Jurusan Teknik Informatika Universitas Islam Negeri (UIN) Alauddin Makassar.
5. Ibu Mustikasari, S.Kom., M.Kom dan Bapak Abdul Wahid, ST., M.Kom selaku Pembimbing I dan Pembimbing II yang dengan keikhlasannya meluangkan waktu, tenaga dan pikirannya dalam pelaksanaan penelitian dan penyusunan skripsi.
6. Bapak Yusran Bobihu, S.Kom., M.Si dan Bapak Hery, S.Kom yang senantiasa memberikan arahan kepada penulis dalam penyusunan skripsi.
7. Kepala Sekolah SMP Negeri 3 Pallangga yang telah memberikan kesempatan kepada penulis untuk melakukan penelitian di sekolah.
8. Dosen-dosen yang mengajar di jurusan Teknik Informatika yang telah banyak memberikan asupan ilmu dari semester awal hingga semester

akhir. Serta Staff Administrasi dan Akademik Fakultas Sains dan Teknologi yang banyak membantu dalam hal persuratan.

9. Spesial kepada Layli Latifah Lestary tercinta beserta keluarga, yang telah memberikan sumbangsih dan selalu memberikan motivasi, semangat serta setia menemani sehingga penyusunan skripsi menjadi lebih indah.
10. Teman-teman seperjuangan Teknik Informatika Angkatan 2007 (ENIAC) yang telah menemani serta memberi canda tawa selama pendidikan dikampus.
11. Teman-teman “D’clacest” dan “Anak Belakang” yang senantiasa memberi penyegaran disaat pikiran lagi suntuk.
12. Seluruh pihak yang tidak dapat penulis sebutkan satu per satu, namun telah banyak terlibat membantu penulis dalam proses penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, maka dari itu dengan segala kerendahan hati penulis bersedia menerima kritik dan saran yang bersifat membangun dari semua pihak, baik dari pendidik maupun pembaca sehingga penulisan skripsi ini akan jauh lebih baik kedepannya.

Pada akhirnya penulis berkhidmat kepada Allah swt., seraya memohon segala usaha hamba bernilai ibadah di sisi-Nya. Amin.

Fastabiqul Khairat, Wabillahi Taufiq Wal Hidayah

Wassalamu’alaikum Wr.Wb.

Makassar, 14 Agustus 2012

Penulis

DAFTAR ISI

	Hal
Halaman Judul	i
Halaman Pernyataan Keaslian Skripsi	ii
Persetujuan Pembimbing	iii
Halaman Pengesahan Skripsi	iv
Kata Pengantar	v
Daftar Isi	viii
Daftar Gambar	x
Daftar Tabel	xi
Abstrak	xii
 BAB I PENDAHULUAN	
A. Latar Belakang	1
B. Rumusan Masalah	6
C. Batasan Masalah	6
D. Pengertian Judul	7
E. Tujuan dan Manfaat Penelitian	8
 BAB II KAJIAN PUSTAKA	
A. Tinjauan Pustaka	9
B. Landasan Teori	10
 BAB III METODE PENELITIAN	
A. Lokasi Penelitian	55
B. Jenis Penelitian	55

	C. Metode Pengumpulan Data	55
	D. Hardware dan Software	56
	E. System Requirement	57
	F. Metode Pengujian Perangkat Lunak	57
BAB IV	ANALISIS DAN DESAIN SISTEM	
	A. Analisis Sistem	59
	B. Deskripsi Umum Perangkat Lunak	62
	C. Perancangan Sistem	63
BAB V	IMPLEMENTASI DAN PENGUJIAN SISTEM	
	A. Implementasi Perangkat Lunak	73
	B. Pengujian Sistem	73
BAB VI	KESIMPULAN DAN SARAN	
	A. Kesimpulan	82
	B. Saran	82
	Daftar Pustaka	83
	Lampiran	
	Riwayat Hidup Penulis	

DAFTAR GAMBAR

	Hal
Gambar II.1 Arsitektur Android	24
Gambar II.2 Hirarki View Android User Interface	33
Gambar IV.1 Use Case Diagram.....	63
Gambar IV.2 Activity Diagram User	64
Gambar IV.3 Activity Diagram Teacher dan Admin	65
Gambar IV.4 Activity Diagram M-Learning.....	66
Gambar IV.5 Activity Diagram E-Learning	67
Gambar IV.6 Sequence Diagram.....	68
Gambar IV.7 Diagram Rancangan Sistem	69
Gambar IV.8 Rancangan Jendela Login	70
Gambar IV.9 Rancangan Halaman Utama	70
Gambar V.1 Jendela Login Screen E-learning.....	73
Gambar V.2 Jendela Halaman Utama E-learning.....	74
Gambar V.3 Jendela Halaman Materi E-learning.....	75
Gambar V.4 Jendela Login Screen M-learning.....	76
Gambar V.5 Jendela Halaman Utama	76
Gambar V.6 Jendela Materi	77
Gambar V.7 Jendela Berita	78
Gambar V.8 Jendela Tugas	78
Gambar V.9 Jendela Nilai	79
Gambar V.10 Jendela Download	80

DAFTAR TABEL

	Hal
Tabel II.1 Simbol Use Case Diagram	49
Tabel II.2 Simbol Sequence Diagram	53
Tabel II.3 Simbol Activity Diagram	54



ABSTRAK

Nama : Sutrisno
NIM : 60200107109
Jurusan : Teknik Informatika
Judul : Aplikasi Pembelajaran Interaktif Untuk Mata Pelajaran Teknologi Informasi Dan Komunikasi Pada SMP Negeri 3 Pallangga

Ilmu pengetahuan dan teknologi merupakan sesuatu yang tidak dapat dipisahkan. Perkembangan ilmu pengetahuan selalu beriringan dengan berkembangnya teknologi. Ilmu pengetahuan adalah hal yang mutlak harus dituntut oleh manusia, agar manusia tidak merugi di kehidupan dunia dan akhirat. Sebagaimana firman Allah Swt. dalam surah Al Balad ayat 8 – 10 bahwa Allah Swt. memberikan dua buah mata, lidah dan dua buah bibir dan telah ditunjukkan kepada manusia dua jalan yaitu jalan kebajikan dan kejahatan. Untuk menjadi manusia yang tidak merugi, maka jalan yang harus ditempuh adalah jalan kebajikan, salah satunya dengan belajar. Proses belajar mengajar merupakan salah satu faktor yang menentukan ilmu pengetahuan tersebut dapat tersalurkan dan dipahami dengan baik oleh manusia khususnya para pelajar yang sedang menuntut ilmu dibangku pendidikan. Saat ini proses belajar mengajar lebih banyak menggunakan cara manual, dengan syarat guru dan siswa harus bertatap muka langsung guna melaksanakan proses belajar mengajar. Ketika guru tidak dapat hadir dalam proses pembelajaran, maka penyaluran ilmu pengetahuan akan terhambat.

Untuk mengatasi hal tersebut, maka dibuatlah sebuah aplikasi pembelajaran yang berbasis *e-learning* dan *m-learning*, agar memudahkan para guru dan siswa saling berinteraksi. Sehingga proses belajar mengajar tidak akan terhambat oleh faktor ketidakhadiran serta faktor waktu dan tempat. Aplikasi ini dapat diakses dengan menggunakan komputer atau laptop serta perangkat *mobile* berbasis Android. Penelitian ini merupakan penelitian kualitatif dengan metode pengumpulan data berupa *library research* dan *field research*.

Dari hasil penelitian ini diperoleh informasi bahwa aplikasi yang dirancang dapat membantu setiap guru dan siswa dalam melaksanakan proses belajar mengajar dengan menggunakan komputer atau *handphone / tablet* berbasis Android untuk mengupload serta mengakses materi-materi pembelajaran khususnya mata pelajaran Teknologi Informasi dan Komunikasi. Dengan penggunaan aplikasi ini diharapkan siswa dengan cepat dapat menyerap dan memahami materi pembelajaran, khususnya mata pelajaran Teknologi Informasi dan Komunikasi. Untuk pengembangan selanjutnya, diharapkan aplikasi *m-learning* ini ditambahkan dengan *fitur-fitur* yang belum ada dan dapat berjalan disemua *platform* sistem operasi *mobile*.

Kata kunci : Aplikasi Pembelajaran, E-learning, M-learning, Sistem Operasi Android

BAB I

PENDAHULUAN

A. Latar Belakang

Perkembangan ilmu pengetahuan yang sedemikian cepatnya memaksa manusia untuk tetap mengupdate dirinya dengan hal baru. Manusia yang memiliki kemampuan untuk berkembang membutuhkan cara – cara yang efektif dan efisien dalam mengupdate kemampuan dirinya. Salah satunya dengan belajar.

Proses belajar dapat dilakukan dengan memanfaatkan teknologi. Teknologi dapat membawa dampak positif berupa kemajuan dan kesejahteraan bagi manusia, artinya bahwa teknologi dapat digunakan untuk pemanfaatan yang sebesar-besarnya (Fakhruddin, 2005 : 25).

Komputer merupakan suatu mesin yang berperan penting dalam proses pembelajaran. Melalui komputer kita dapat membuat suatu modul pembelajaran sehingga dengan adanya modul ini orang – orang yang ingin belajar tetapi memiliki keterbatasan waktu yang tidak memungkinkan untuk mengikuti kegiatan pembelajaran dapat belajar di waktu yang diinginkannya (Hendranto, 2006 : 6).

Bidang pendidikan mendapatkan dampak yang cukup berarti dengan perkembangan teknologi ini. Pada dasarnya pendidikan merupakan suatu proses komunikasi dan informasi dari pendidik kepada peserta didik yang berisi informasi-informasi pendidikan, yang memiliki unsur-unsur pendidik sebagai sumber informasi, media sebagai penyajian ide, gagasan dan materi pendidikan serta peserta didik itu

sendiri, beberapa bagian unsur ini mendapatkan sentuhan media teknologi informasi, sehingga mencetuskan lahirnya ide tentang *e-learning* (Muhammad Adri, 2008 : 1).

Namun dewasa ini peran *e-learning* mulai tergeser dengan adanya *m-learning* (*mobile learning*). Dalam hal ini *m-learning* tidak pernah menggantikan *e-learning*, karena sarana yang digunakan memang berbeda. Jika *e-learning* menggunakan komputer PC atau laptop sebagai sarana dalam proses pembelajaran, maka *m-learning* mengacu pada teknologi *mobile* (*handphone*) sebagai sarana dalam proses pembelajaran.

Saat ini belum banyak lembaga pendidikan yang memanfaatkan *m-learning* dalam proses pembelajaran. Tetapi melihat perkembangan *mobile* yang sangat cepat, teknologi *m-learning* akan semakin banyak digunakan karena melihat kebutuhan pengguna yang saat ini lebih banyak menggunakan teknologi *mobile*. Selain itu, teknologi *m-learning* memiliki keunggulan dalam hal kepraktisannya, karena media yang digunakan cukup kecil dan bisa dibawa kemana-mana. Namun konsekuensinya konten-konten yang bisa digunakan dalam *m-learning* harus memiliki ukuran file yang relatif kecil.

Mobile learning merupakan paradigma pembelajaran memanfaatkan teknologi dan perangkat *mobile* yang diperkirakan akan mengalami perkembangan pesat dan potensial seiring dengan perkembangan teknologi *mobile* itu sendiri. Hal ini dapat dilihat dari data statistik bahwa dari 240 juta jumlah penduduk Indonesia, 45-50 juta di antaranya adalah pengguna/konsumen seluler. Sedangkan secara teknis, perangkat *mobile* yang beredar saat ini sebenarnya telah memiliki kapabilitas untuk

menjalankan konten-konten berupa multimedia. Multimedia adalah penyampaian informasi menggunakan gabungan dari teks, grafik, suara, video dan animasi (Chomsin S. Widodo, 2008 : 26).

Tidak jauh berbeda dengan *e-learning*, *m-learning* juga memungkinkan penggunaan konten multimedia. Dengan adanya pemanfaatan multimedia dibidang pendidikan, maka cara-cara pembelajaran di dunia pendidikan semakin beraneka ragam. Lembaga riset dan penerbitan komputer yaitu *Computer Technology Research* (CTR) menyatakan bahwa :

”Orang hanya mampu mengingat 20 % dari yang dilihat dan 30 % dari yang didengar. Akan tetapi orang mengingat 50 % dari yang dilihat dan didengar dan 80 % dari yang dilihat, didengar dan dilakukan sekaligus” (Ansar, 2011).

Sesuai dengan hasil riset di atas, dapat disimpulkan bahwa pembelajaran dengan menggunakan multimedia akan dapat mempercepat penyerapan materi oleh peserta didik.

Firman Allah Swt. dalam surah Al Balad ayat 8-10 :

أَلَمْ نَجْعَلْ لَهُ عَيْنَيْنِ ۖ وَلِسَانًا وَشَفَتَيْنِ ۚ وَهَدَيْنَاهُ النَّجْدَيْنِ ﴿٨﴾

UNIVERSITAS ISLAM NEGERI

ALAUDDIN
M A K A S S A R

Terjemahnya :

“Bukankah kami Telah memberikan kepadanya dua buah mata, lidah dan dua buah bibir. Dan kami Telah menunjukkan kepadanya dua jalan yaitu jalan kebajikan dan jalan kejahatan.” (Depag, 1971 : 816).

Rasulullah Saw. bersabda pada hadis kursi: "Allah berfirman: Hai anak Adam Aku telah memberimu nikmat yang amat banyak, engkau tidak akan dapat menghitungnya dan mensyukurinya. Dua mata untuk engkau melihat dengannya dan aku beri tutup maka gunakan untuk melihat apa yang Aku halalkan dan apabila engkau melihat benda yang Aku haramkan maka tutuplah matamu. Juga Aku beri kepada mu lidah dan Aku beri alat penutupnya, maka gunakanlah untuk mengucapkan apa yang Aku perintahkan dan halalkan bagimu, jika ada sesuatu yang aku haramkan maka tutuplah rapat-rapat lidahmu, juga Aku berikan kepada kamu kemaluan dan Aku beri tutup, maka pergunakanlah kemaluanmu itu untuk apa yang Aku halalkan dan jika ada sesuatu yang Aku haramkan maka tutuplah. hai anak Adam engkau tidak sanggup menanggung murka Ku dan tidak kuat menerima siksa pembalasan Ku".

Pada Ayat tersebut menjelaskan bahwa Allah Swt. memberikan manusia dua buah mata, lidah dan dua buah bibir. Yang lebih baik digunakan kepada jalan kebajikan dan tidak untuk jalan kejahatan. Ayat ini pun terkait dengan firman Allah Swt. dalam surah Al Israa' ayat 36 :

وَلَا تَقْفُ مَا لَيْسَ لَكَ بِهِ عِلْمٌ إِنَّ السَّمْعَ وَالْبَصَرَ وَالْفُؤَادَ كُلُّ أُولَٰئِكَ كَانَ عَنْهُ

UNIVERSITAS ISLAM NEGERI

ALAUDDIN
MAKASSAR

مَسْئُولًا

Terjemahnya :

“Dan janganlah kamu mengikuti apa yang kamu tidak mempunyai pengetahuan tentangnya. Sesungguhnya pendengaran, penglihatan dan hati, semuanya itu akan diminta pertanggungan jawabnya.” (Depag, 1971 : 429).

Ayat di atas, penulis ingin menselaraskan antara Islam dan teknologi. Dengan cara menyeimbangkan indera penglihatan dan indera pendengaran untuk satu tujuan, yaitu belajar. Penjelasan program-program aplikasi bisnis seperti Microsoft Word, Excel dan Jaringan komputer adalah program yang digunakan pada Sekolah Menengah Pertama sebagai bahan ajar dari mata pelajaran Teknologi Informasi dan Komunikasi. Salah satu sekolah yang mengajarkan mata pelajaran Teknologi Informasi dan Komunikasi adalah SMP Negeri 3 Pallangga yang terletak di Kabupaten Gowa, Provinsi Sulawesi Selatan.

Saat ini proses belajar mengajar khususnya pada SMP Negeri 3 Pallangga dilakukan dengan cara manual yaitu, guru menjelaskan materi kepada peserta didik tanpa ada media *visualisasi* yang diperlihatkan, sehingga banyak materi yang disampaikan oleh guru tidak dapat diserap dengan baik oleh peserta didik. Faktor penghambat lainnya bahwa tenaga pengajar tidak selamanya dapat hadir dalam setiap proses belajar mengajar karena sakit atau lain hal, maka untuk memperlancar proses pembelajaran, teknologi seperti *e-learning* dan *m-learning* merupakan *alternatif* yang sangat tepat.

Bentuk dari *e-learning* itu sendiri bermacam-macam, salah satunya adalah implementasi *e-learning* dalam bentuk moodle. Sedangkan bentuk *m-learning* pula bermacam-macam. Untuk itu diperlukan aplikasi yang dapat menselaraskan antara *e-learning* dan *m-learning*, aplikasi ini dapat pula disebut dengan aplikasi *client-server*. Dimana *client* itu berupa *m-learning* yang dapat diakses dengan menggunakan *device* yang bersistem operasi android, dan *server* itu berupa *e-learning* (moodle) yang dapat diakses menggunakan *device* laptop ataupun Personal Computer (PC).

Dengan memanfaatkan perkembangan teknologi komputer dan *mobile* pada bidang pendidikan, tentu akan membawa suasana baru dan dampak positif dalam proses pembelajaran.

Berdasarkan dengan latar belakang masalah tersebut maka penulis membuat Tugas akhir ini dengan judul **"Aplikasi Pembelajaran Interaktif Untuk Mata Pelajaran Teknologi Informasi & Komunikasi Pada SMP Negeri 3 Pallangga"**

B. Rumusan Masalah

Dari latar belakang di atas, maka yang menjadi rumusan masalahnya adalah "bagaimana cara perancangan aplikasi pembelajaran interaktif untuk mata pelajaran Teknologi Informasi & Komunikasi dengan berbasis *e-learning* (moodle) dan *mobile learning* (android) pada SMP Negeri 3 Pallangga yang disesuaikan dengan kurikulum yang ada".

C. *Batasan Masalah*

Adapun batasan masalahnya adalah :

1. Pembuatan materi sebagai bahan ajar untuk mata pelajaran Teknologi Informasi dan Komunikasi disesuaikan dengan kurikulum SMP yang berlaku.
2. Aplikasi *e-learning* (moodle) yang dibuat dapat berjalan pada berbagai sistem operasi komputer dan bersinergi dengan aplikasi *m-learning* yang dibuat khusus untuk perangkat *mobile* yang berbasis android OS.

D. *Pengertian Judul*

Untuk menghindari salah pengertian judul ataupun makna yang terkandung pada judul skripsi ini, maka dijelaskan beberapa istilah dalam judul skripsi ini.

1. Aplikasi adalah software program yang memiliki aktivitas pemrosesan perintah yang diperlukan untuk melaksanakan permintaan pengguna dengan tujuan tertentu (Febrian Jack, 2010 : 256).
2. Pembelajaran adalah perubahan yang diperlihatkan dalam perubahan tingkah laku, yang keadaannya berbeda dari sebelum individu berada dalam situasi belajar dan sesudah melakukan tindakan yang serupa itu. Perubahan terjadi akibat adanya suatu pengalaman atau latihan, berbeda dengan perubahan serta-merta akibat refleksi atau perilaku yang bersifat naluriah (Nasution, 2010 : 7).

3. Interaktif adalah proses yang lebih tertuju pada penalaran dan pemecahan serta adanya bentuk komunikasi langsung (tatap muka) dan mengontrol penuh apa yang diinginkan sehingga terjalin interaksi secara intensif (Binanto Iwan, 2010 : 2).
4. Teknologi Informasi dan Komunikasi adalah ilmu pengetahuan yang lebih mendalami tentang perkembangan informasi dan komunikasi serta perangkat-perangkat yang turut berperan serta dalam mewujudkan masyarakat modern (Febrian, 2010 : 73).

E. Tujuan dan Manfaat Penelitian

1. Tujuan Penelitian

Untuk membuat aplikasi pembelajaran interaktif yang disesuaikan dengan kurikulum pada SMP Negeri 3 Pallangga.

2. Manfaat Penelitian

Adapun manfaat yang diharapkan dari penelitian ini adalah :

1. Dapat dijadikan sebagai media pembelajaran untuk mata pelajaran Teknologi Informasi dan Komunikasi pada tingkat Sekolah Menengah Pertama
2. Sistem ini dirancang khusus untuk membantu para guru Teknologi Informasi dan Komunikasi dalam pelaksanaan kegiatan proses belajar mengajar.

BAB II

KAJIAN PUSTAKA

A. Tinjauan Pustaka

Banyak penelitian yang telah dilakukan yang mengembangkan media pembelajaran berbasis *mobile* (*mobile learning*). Penelitian tersebut di antaranya pembuatan media pembelajaran bahasa Inggris dengan *mobile learning* yang dikembangkan oleh Anararki dari Universitas Assumption, Bangkok Thailand pada tahun 2009 (Firouz Anararki, 2009). Pada penelitian tersebut dikembangkan *mobile learning* dengan menggunakan animasi flash atau yang dikenal dengan Flash Mobile Learning. Penelitian ini dikembangkan untuk pembelajaran bahasa Inggris bagi pengguna bahasa lokal di Thailand.

Penelitian lain yang terkait adalah penelitian yang dilakukan oleh Salemah yang mengembangkan sistem multimedia *offline* telepon seluler untuk pembelajaran bahasa Inggris (Osama Salameh : 2011). Selain menggunakan flash for mobile (animasi flash untuk telepon seluler) media pembelajaran dapat dikembangkan dengan menggunakan *Java mobile* (J2ME). Java dipilih karena saat ini berbagai mobile phone sudah memiliki fitur untuk menjalankan aplikasi berbasis java.

Melihat dari beberapa penelitian yang ada, belum ada yang membahas tentang pembuatan aplikasi pembelajaran interaktif khususnya untuk mata pelajaran Teknologi Informasi & Komunikasi yang mengkombinasikan antara *e-learning* (moodle) dengan *m-learning* yang berbasis Android.

B. Landasan Teori

1. Defenisi Multimedia

Istilah multimedia berasal dari teater, bukan komputer. Pertunjukan yang menggunakan lebih dari satu medium seringkali disebut dengan pertunjukan multimedia. Sistem multimedia dimulai pada akhir tahun 1980-an dengan diperkenalkannya *Hypercard* oleh Apple pada tahun 1987 dan pengumuman oleh IBM pada tahun 1989 mengenai perangkat lunak *audio visual connection (AVC)* dan *video adapter card* (Purwanto Agus, 2008 : 2).

Secara umum, multimedia berhubungan dengan pengguna lebih dari satu macam cara untuk menyajikan informasi. Rekaman musik hanya menggunakan suara (mungkin disebut *unimedia*), musik video adalah bentuk multimedia karena informasi menggunakan suara dan video.

Secara teknis, multimedia dalam komputer bukan hal yang baru, karena integrasi antara teks dan gambar merupakan komponen utama dalam banyak perangkat lunak aplikasi yang sudah lama dilakukan. Pada saat ini, multimedia mempunyai arti tidak hanya integrasi antara teks dan grafik sederhana saja, tetapi dilengkapi dengan suara dan animasi. Sambil mendengarkan penjelasan, dapat melihat gambar, animasi dan membaca penjelasan dalam bentuk teks. Multimedia terbagi menjadi 2 macam, yaitu multimedia linear dan multimedia interaktif (Sigit Bambang, 2011 : 13).

2. Komponen Multimedia

Menurut James A. Senn dalam buku Agus Purwanto, multimedia terbagi menjadi beberapa elemen-elemen yaitu :

a. Teks

Hampir semua orang yang dapat menggunakan komputer sudah terbiasa dengan teks. Teks merupakan dasar dari pengolahan kata dan informasi yang berbasis multimedia. Dalam kenyataannya, multimedia menyajikan informasi kepada audiens dengan cepat, karena tidak diperlukan membaca secara rinci dan teliti.

b. Grafik

Secara umum, grafik berarti gambar atau garis (*line drawing*). Manusia sangat berorientasi pada visual (*visual oriented*), dan gambar merupakan sarana yang sangat baik untuk menyajikan informasi. Grafik merupakan komponen penting dalam multimedia.

c. Image

Secara umum image berarti gambar atau *raster* (*halfone drawing*), seperti foto. Basis data karyawan dengan atribut seperti nama, alamat dan lainnya lebih efektif bila foto karyawan yang bersangkutan dapat ditampilkan. Demikian juga foto-foto seperti gedung dan lain - lain sangat memerlukan penyimpanan yang besar. Hal inilah yang menyebabkan aplikasi multimedia disimpan dalam penyimpanan yang cukup besar kapasitasnya seperti *CD-Room*.

d. Animasi

Animasi berarti gerakan image atau video, contohnya gerakan orang yang melakukan suatu kegiatan. Konsep dari animasi adalah sulitnya menggambarkan informasi dengan satu gambar saja, atau sekumpulan gambar, juga tidak dapat menggunakan teks untuk menerangkan informasi. Arsip animasi memerlukan penyimpanan yang jauh lebih besar dari pada satu gambar.

e. Suara

Suara dapat lebih memperjelas pengertian yang ditampilkan dalam teks dan video. Contohnya, narasi merupakan kelengkapan dari penjelasan yang dilihat melalui video. Suara dapat memberikan lebih menjelaskan karakteristik suatu gambar, misalnya musik dan suara efek (sound effect).

f. Interaktif Link

Sebagian dari multimedia adalah interaktif, dimana pengguna dapat menekan mouse atau objek pada screen seperti button atau teks dan menyebabkan program melakukan perintah tertentu.

Interaktif link dengan informasi yang dihubungkannya sering kali dihubungkan secara keseluruhan sebagai *hypermedia*. Secara spesifik, dalam hal ini termasuk *hypertext* (*hotword*), *hypergraphics* dan *hypersound* menjelaskan jenis informasi yang dihubungkan.

Interaktif link diperlukan jika pengguna, menunjuk pada suatu objek atau button agar dapat mengakses program tertentu. Interaktif link diperlukan untuk menggabungkan beberapa elemen multimedia sehingga menjadi informasi yang terpadu. Cara pengaksesan informasi pada multimedia terdapat dua macam yaitu linier dan non-linier. Informasi linier adalah informasi yang ditampilkan secara sekuensial, yaitu dari atas ke bawah atau halaman demi halaman. Sedangkan pada informasi non-linier informasi dapat ditampilkan langsung sesuai dengan kehendak pengguna.

3. Konsep Pembelajaran Interaktif

Pemrograman interaktif merupakan pandangan atau filosofi dalam proses pembuatan program yang menunjukkan bahwa hasil program dioperasikan secara interaktif dengan mudah, jelas dan sederhana bagi pemakai. Pemahaman pemrograman yang interaktif terdiri dari dua faktor utama, yaitu fasilitas antar muka pemakai (*user interface*) dan sistem komputer yang digunakan oleh pemakai secara langsung. Sedangkan interaksi antara pemakai dan komputer meliputi semua aspek yang berhubungan dengan cara atau metode bagaimana pemakai berinteraksi dengan komputer.

Desain antarmuka pemakai dengan pengontrolan yang terlihat jelas dan hasil tampilan menu program menunjukkan fungsi-fungsi yang jelas bagi pemakai sistem. Interaksi pemakai dan komputer (program) memperlihatkan rancangan, desain dan

implementasi program menunjukkan kemudahan dalam operasional dan mudah dipelajari oleh orang awam.

Tampilan program sistem secara grafis, secara ideal akan membantu pemahaman pemakai dalam mempergunakan/mengoperasikan sistem. Beberapa karakteristik yang mempengaruhi dalam mendesain tampilan program sistem dengan desain grafis yang terstruktur dan baik yaitu :

1. Bentuk penampilan, struktur secara fisik dan hubungan antar bagian secara keseluruhan.
2. Penempatan posisi, gerakan fisik menu, alur sistem, dan proses interaksi yang terjadi terhadap suatu hubungan interaktif
3. Ukuran bentuk tampilan, jumlah tampilan per modul menu, kecendrungan penggunaan, pengelompokkan fungsi menu yang serasi dan seragam.
4. Penempatan posisi atau pemetaan menu disesuaikan dengan objek yang serasi, dengan mempertimbangkan objek lain dalam program sistem.

Faktor penting lain yang mempengaruhi hasil rancangan sistem yang interaktif dan kemudahan dalam penggunaan, adalah faktor pewarnaan untuk memberikan hasil pandangan atau visualisasi program yang lembut dan memberikan kenyamanan bagi mata pemakai. Faktor utama yang mempengaruhi bentuk warna pada tampilan menu program yaitu segmentasi warna menu pada menu untuk membedakan antar bagian, misalnya dengan mengkombinasikan warna halus/gelap untuk dasar dan warna. terang untuk menu pilihan (Kasim, 2009 : 13).

4. Sistem Operasi Android

Pada tahun 2005 Google mengakuisisi Android Inc. yang pada saat itu dimotori oleh Andy Rubin, Rich Miner, Nick Sears dan Chris White. Yang kemudian pada tahun itu juga memulai membangun platform Android secara intensif (Mulyadi, 2010 : 2).

Android adalah sebuah sistem operasi untuk perangkat mobile berbasis linux yang mencakup system operasi, *middleware* dan aplikasi. Android menyediakan platform yang terbuka bagi para pengembang untuk menciptakan aplikasi mereka (Nazruddin, 2011 : 1).

Operating System Android ini di keluarkan oleh Google untuk menyaingi Windows, Apple dan pesaing-pesaingnya. Google sendiri mempunyai alasan cukup kuat untuk melirik pangsa seluler, karena perkembangan teknologi telepon seluler dewasa ini sudah bukan merupakan evolusi lagi, melainkan sebuah revolusi. Babak baru dalam dunia telekomunikasi nirkabel ini terus bergulir dengan cepat.

Kemudian untuk mengembangkan Android, dibentuklah Open Handset Alliance, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia. Secara garis besar, arsitektur Android terdiri atas *Applications* dan *Widgets*, *Applications Frameworks*, *Libraries*, *Android Run Time*, dan *Linux Kernel* (Supardi Yuniar, 2011 : 4).

Pesatnya pertumbuhan Android yaitu platform yang sangat lengkap baik itu system operasinya, Aplikasi dan Tool Pengembangan, Market aplikasi android serta dukungan yang sangat tinggi dari komunitas Open Source di dunia, sehingga android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah device yang ada di dunia.

Salah satu hal yang membuat perangkat Android menarik adalah tersedianya banyak aplikasi yang bisa dipakai oleh penggunanya. Berdasarkan data terakhir pada saat event Google IO 2011, tercatat sudah ada 200.00 aplikasi yang tersedia di Android Market. Jumlah tersebut tidaklah sedikit.

5. The Dalvik Virtual Machine (DVM)

Salah satu elemen kunci dari android adalah Dalvik Virtual Machine (DVM). Android berjalan di dalam Dalvik Virtual Machine (DVM) bukan di Java Virtual Machine (JVM), sebenarnya banyak persamaannya dengan java virtual machine seperti Java ME (Java Mobile Edition), tetapi Android menggunakan Virtual Machine sendiri dikostominasi dan dirancang untuk memastikan bahwa beberapa *feature-feature* berjalan lebih efisien pada perangkat *mobile*.

Dalvik Virtual Machine (DVM) adalah “register bases” sementara Java Virtual Machine (JVM) adalah “stack bases”, DVM didesain dan ditulis oleh Dan Bornsten dan beberapa engineeres Goole lainnya. Jadi bisa kita katakan “Dalvik equals(Java) == False”. Dalvik Virtual Machine menggunakan kernel Linux untuk menangani fungsionalitas. tingkat rendah termasuk keamanan, threading dan proses

serta manajemen memori. Ini memungkinkan kita untuk menulis aplikasi C / C++ sama halnya seperti pada OS Linux kebanyakan. Meskipun dalam kenyataannya kita harus banyak memahami Arsitektur dan proses sistem dari kernel linux yang digunakan dalam Android tersebut.

Semua hardware yang berbasis Android dijalankan dengan menggunakan Virtual Machine untuk eksekusi aplikasi, pengembang tidak perlu khawatir tentang implementasi perangkat keras tertentu. *Dalvik Virtual Machine* mengeksekusi *executable file*, sebuah format yang dioptimalkan untuk memastikan memori yang digunakan sangat kecil. *The executable file* diciptakan dengan mengubah kelas bahasa Java dan dikompilasi menggunakan *tools* yang disediakan dalam SDK Android.

6. Android SDK (*Software Development Kit*)

Android SDK adalah tools API (Application Programming Interface) yang diperlukan untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Android merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, middleware dan aplikasi kunci yang direlease oleh Google. Saat ini disediakan Android SDK (Software Development Kit) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi-netral, Android memberi kesempatan untuk membuat aplikasi yang kita butuhkan yang bukan merupakan aplikasi bawahan handphone/smartphone.

Beberapa fitur-fitur android yang paling penting adalah :

- a. Framework Aplikasi yang mendukung penggantian komponen dan reusable.
- b. Mesin Virtual Dalvik dioptimalkan untuk perangkat mobile.
- c. Integrated browser berdasarkan engine open source WebKit.
- d. Grafis yang dioptimalkan dan didukung oleh libraries grafis 2D, grafis 3D berdasarkan spesifikasi opengl ES 1.0 (Opsional akselerasi hardware).
- e. SQLite untuk penyimpanan data (database)
- f. Media Support yang mendukung audio, video, dan gambar (MPEG4, H.264, MP3, AAC, JPG, PNG, GIF), GSM Telephony (tergantung hardware).
- g. Bluetooth, EDGE, 3G dan WiFi (tergantung hardware)
- h. Kamera, GPS, kompas, dan *accelerometer* (tergantung hardware)
- i. Lingkungan *Development* yang lengkap dan kaya termasuk perangkat emulator, tools untuk debugging, profil dan kinerja memori dan plugins untuk IDE Eclipse.

7. Arsitektur Android

Secara garis besar Arsitektur Android dapat dijelaskan dan digambarkan sebagai berikut :

a. Application dan Widgets

Application dan Widgets ini adalah layer dimana kita berhubungan dengan aplikasi saja, dimana biasanya kita download aplikasi kemudian kita lakukan instalasi dan jalankan aplikasi tersebut. Di layer terdapat aplikasi inti termasuk klien email, program SMS, kalender, peta, browser, kontak, dan lain-lain. Semua aplikasi ditulis menggunakan bahasa pemrograman Java.

b. Applications Frameworks

Android adalah “ Open Development Platform” yaitu Android menawarkan kepada pengembang atau member kemampuan kepada pengembang untuk membangun aplikasi yang bagus dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi resources, menjalankan service background, mengatur alarm, dan menambahkan status notification, dan sebagainya. Pengembang memiliki akses penuh menuju API framework seperti yang dilakukan oleh aplikasi yang kategori inti. Arsitektur aplikasi dirancang supaya kita dengan mudah dapat menggunakan kembali komponen yang sudah digunakan.

Jadi, Application framework ini adalah layer dimana para pembuat aplikasi melakukan pengembangan/pembuatan aplikasi yang akan dijalankan di sistem operasi android, karena pada layer inilah aplikasi dapat dirancang dan dibuat, seperti content providers yang berupa sms dan panggilan telepon.

Komponen-komponen yang termasuk di dalam Application frameworks adalah sebagai berikut :

1. Satu set Views : yang dapat digunakan untuk membangun aplikasi meliputi list, grid, kotak teks, tombol, dan sebuah embeddable web.
2. Content Provider : yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain (misalnya kontak telepon), atau untuk membagi data tersebut.
3. Resource Manager : menyediakan akses ke kode non sumber lokal misalnya string, gambar, dan tata letak file.
4. Notification Manager : yang memungkinkan semua aplikasi menampilkan alert yang bisa dikustomisasi di dalam status bar.
5. Activity Manager : yang mengatur life cycle dari aplikasi dan menyediakan navigasi umum backstack (Mulyadi, 2011 : 8)

c. Libraries

Lapisan setelah Kernel Linux adalah *Libraries*. Android menyertakan satu set *library-library* dalam bahasa C/C++ yang digunakan oleh berbagai komponen yang ada pada sistem Android. Kemampuan ini dapat diakses oleh programmer melewati Android *application framework*. Sebagai contoh Android mendukung pemutaran format audio, video, dan gambar.

Berikut ini beberapa *core library* tersebut :

1. System C library : variasi dari implementasi BSD, berdasar pada sistem standar C library (libc) yang sesuai untuk perangkat berbasis Linux.
2. Media Libraries : Paket Video berdasarkan OpenCORE; *library* mendukung pemutaran rekaman dan banyak format populer audio dan video, meliputi MPEG4, H.264, MP3, AAC, AMR, JPG, dan PNG.
3. Surface Manager : mengatur akses ke subsistem layar, lapisan komposit 2D and grafis 3D dari berbagai aplikasi.
4. LibWebCore : mesin web browser modern yang powerfull.
5. SGL : mesin grafis 2D.
6. 3D libraries : implementasi berdasarkan OpenGL ES 1.0 APIs; *library* ini menggunakan *hardware 3D acceleration* dan *highly optimized 3D software rasterizer*.

7. FreeType : bitmap dan vektor font *rendering*.
8. SQLite : mesin database yang powerful dan ringan tersedia untuk semua aplikasi. (Reto, 2010 : 13)

d. Android Run Time

Layer yang membuat aplikasi Android dapat dijalankan dimana dalam prosesnya menggunakan Implementasi linux. Bertempat di level yang sama dengan Libraries adalah Android-Runtime. Android Runtime ini berisi Core Libraries dan Dalvik Virtual Machine.

Core Libraries mencakup serangkaian inti library Java, artinya Android menyertakan satu set library-library dasar yang menyediakan sebagian besar fungsi-fungsi yang ada pada library-library dasar bahasa pemrograman Java.

Dalvik adalah Java Virtual Machine yang memberi kekuatan pada sistem Android. Dalvik Virtual Machine ini di optimalkan untuk telepon seluler.

Setiap aplikasi yang berjalan pada Android berjalan pada prosesnya sendiri, dengan instance dari Dalvik Virtual Machine. Dalvik telah dibuat sehingga sebuah piranti yang memakainya dapat menjalankan multi Virtual Machine dengan efisien. Dalvik Virtual Machine dapat mengeksekusi file dengan format Dalvik Executable (.dex) yang telah dioptimasi untuk menggunakan memory seminimal mungkin. Virtual Machine ini register-based, dan menjalankan class-class

yang di-compile menggunakan compiler Java yang kemudian ditransformasi menjadi format .dex menggunakan "dx" tool yang telah disertakan.

Dalvik Virtual Machine (DVM) menggunakan kernel Linux untuk menjalankan fungsi-fungsi seperti threading dan low-level memory management.

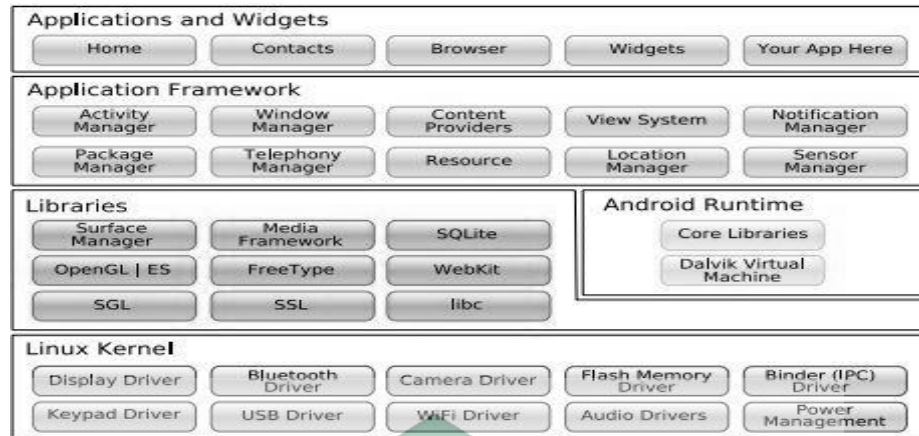
Perbedaan antara Dalvik VM dan Java murni adalah :

- 1) Dalvik VM menjalankan file .dex yang merupakan hasil konversi pada saat kompilasi program dari .class dan .jar. File .dex lebih padat dan efisien dari pada file .class. Hal ini akan menghemat daya dan mengurangi konsumsi memory.
- 2) Core Java Libraries yang ada di Android itu berbeda dengan Java SE maupun Java ME. (Ed Burnette, Hello Android, Introducing Google's Mobile Application Development), edisi 3 (United States of America : Pragmatic Programmers LLC, 2010 : 33).

e. Linux Kernel

Linux Kernel adalah layer dimana inti dari operating system dari Android itu berada. Berisi file-file system yang mengatur system processing, memory, resource, drivers, dan system-sistem operasi

android lainnya. Linux Kernel yang digunakan android adalah linux kernel release 2.6



Gambar II.1. Arsitektur Android

Sumber : Nazruddin Safaat, Android Pemrograman Aplikasi Mobile Smartphone dan Tablet Berbasis Android, 2011.

8. Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman java. Kode java dikompilasi bersama dengan data file secara resource yang dibutuhkan oleh aplikasi, dimana prosesnya di-package oleh tools yang dinamakan ‘apt tools’ ke dalam paket android sehingga menghasilkan file dengan ekstensi apk. File apk itulah yang kita sebut dengan aplikasi dan nanti dapat diinstall di perangkat mobile.

Beberapa jenis komponen pada aplikasi Android, yaitu :

(a) Activities

Activity merupakan bagian yang paling penting dalam sebuah aplikasi, karena *Activity* menyajikan tampilan visual program yang sedang digunakan oleh pengguna. Setiap *Activity* dideklarasikan dalam sebuah kelas yang bertugas untuk menampilkan antarmuka pengguna yang terdiri dari *Views* dan respon terhadap *Event*. Setiap aplikasi memiliki sebuah *activity* atau lebih. Biasanya pasti akan ada *activity* yang pertama kali tampil ketika aplikasi dijalankan. Perpindahan antara *activity* dengan *activity* lainnya diatur melalui sistem, dengan memanfaatkan *activity stack*. Keadaan suatu *activity* ditentukan oleh posisinya dalam tumpukan *activity*, LIFO (*Last In First Out*) dari semua aplikasi yang sedang berjalan.

Bila suatu *activity* baru dimulai, *activity* yang sebelumnya digunakan maka akan dipindahkan ketumpukan paling atas. Jika pengguna ingin menggunakan *activity* sebelumnya, cukup menekan tombol *Back*, atau menutup *activity* yang sedang digunakan, maka *activity* yang berada diatas akan aktif kembali. *Memory Manager* Android menggunakan tumpukkan ini untuk menentukan prioritas aplikasi berdasarkan *activity*, memutuskan untuk mengakhiri suatu aplikasi dan mengambil sumber daya dari aplikasi tersebut.

Ketika *activity* diambil dan disimpan dalam tumpukan *activity* terdapat 4 kemungkinan kondisi transisi yang akan terjadi, yaitu :

- 1) **Active**, setiap *activity* yang berada ditumpukan paling atas, maka dia akan terlihat, terfokus, dan menerima masukkan dari pengguna. Android akan berusaha untuk membuat *activity* aplikasi ini untuk tetap hidup dengan segala cara, bahkan akan menghentikan *activity* yang berada dibawah tumpukkannya jika diperlukan. Ketika *activity* sedang aktif, maka yang lainnya akan dihentikan sementara.
- 2) **Paused**, dalam beberapa kasus *activity* akan terlihat tapi tidak terfokus pada kondisi inilah disebut *paused*. Keadaan ini terjadi jika *activity* transparan dan tidak *fullscreen* pada layar. Ketika *activity* dalam keadaan *paused*, dia terlihat *active* namun tidak dapat menerima masukkan dari pengguna. Dalam kasus ekstrim, Android akan menghentikan *activity* dalam keadaan *paused* ini, untuk menunjang sumber daya bagi *activity* yang sedang aktif.
- 3) **Stopped**, ketika sebuah *activity* tidak terlihat, maka itulah yang disebut *stopped*. *Activity* akan tetap berada dalam memori dengan semua keadaan dan informasi yang ada. Namun akan menjadi kandidat utama untuk dieksekusi oleh sistem ketika membutuhkan sumberdaya lebih. Oleh karenanya ketika suatu *activity* dalam

kondisi *stopped* maka perlu disimpan data dan kondisi antarmuka saat itu. Karena ketika *activity* telah keluar atau ditutup, maka dia akan menjadi *inactive*.

- 4) **Inactive**, kondisi ketika *activity* telah dihentikan dan sebelum dijalankan. *Inactive activity* telah diiadakan dari tumpukan *activity* sehingga perlu *restart* ulang agar dapat tampil dan digunakan kembali. Kondisi transisi ini sepenuhnya ditangani oleh manajer memori Android. Android akan memulai menutup aplikasi yang mengandung *activity inactive*, kemudian *stopped activity*, dan dalam kasus luar biasa *paused activity* juga akan di tutup.

(b) Service

Suatu *service* tidak memiliki tampilan antarmuka, melainkan berjalan di *background* untuk waktu yang tidak terbatas. Komponen *service* diproses tidak terlihat, memperbarui sumber data dan menampilkan notifikasi. *Service* digunakan untuk melakukan pengolahan data yang perlu terus diproses, bahkan ketika *Activity* tidak aktif atau tidak tampak.

(c) Broadcast Receiver

Broadcast Receiver merupakan komponen yang sebenarnya tidak melakukan apa-apa.

(d) Content Provider

Content Provider yaitu mekanisme yang memungkinkan sebuah aplikasi berbagi informasi dengan aplikasi lainnya. (Murphy, 2010 : 3).

9. Versi Android

Sejak diluncurkan pada tanggal 23 September 2008, Android telah mengalami beberapa kali pengembangan dan perbaikan terhadap versi sistem operasinya. Yang menarik dari versi-versi sistem operasi Android ini ialah penggunaan codename untuk setiap versi yang mengambil nama hidangan penutup dan memiliki pola keterurutan nama berdasarkan abjad.

Versi yang pertama adalah Donut (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus, kamera, camcorder dan galeri yang dintegrasikan, CDMA / EVDO, 802.1x, VPN, *Gestures*, dan *Text-to-speech engine*, kemampuan dial kontak, teknologi *text to change speech* (tidak tersedia pada semua ponsel), pengadaan resolusi VWGA.

Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (Eclair), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan *flash* untuk kamera 3,2 MP, digital zoom, dan Bluetooth 2.1.

Pada 20 Mei 2010, Android versi 2.2 (Froyo) diluncurkan. Perubahan-perubahan umumnya terhadap versi-versi sebelumnya antara lain dukungan Adobe Flash 10.1, kecepatan kinerja dan aplikasi 2 sampai 5 kali lebih cepat, integrasi V8 JavaScript *engine* yang dipakai Google Chrome yang mempercepat kemampuan rendering pada browser, pemasangan aplikasi dalam SD Card, kemampuan Wi-Fi Hotspot portabel, dan kemampuan *auto update* dalam aplikasi Android Market.

Pada 6 Desember 2010, Android versi 2.3 (Gingerbread) diluncurkan. Perubahan-perubahan umum yang didapat dari Android versi ini antara lain peningkatan kemampuan permainan (*gaming*), peningkatan fungsi copy paste, layar antar muka (*User Interface*) didesain ulang, dukungan format video VP8 dan WebM, efek audio baru (*reverb*, *equalization*, *headphone virtualization*, dan *bass boost*), dukungan kemampuan *Near Field Communication* (NFC), dan dukungan jumlah kamera yang lebih dari satu (Safaat, 2011 : 11-12).

Android versi 3.0 (Honeycomb) diluncurkan pada Juli 2011. Tidak butuh waktu yang lama, Google kembali meluncurkan Android versi 4.0 yang diberi nama versi ICS atau Ice Cream Sandwich pada November 2011. Android versi 3.0 keatas adalah generasi *platform* yang digunakan untuk tablet pc. Sementara Android versi 4.0 ini sudah merupakan *platform* yang bisa dipakai si *smartphone* dan tablet pc.

10. User Interface

Android *User Interface* (UI) bisa dibuat dengan dua cara, yaitu dengan menggunakan kode XML atau menulisnya melalui kode Java. Membuatnya dengan XML sangat direkomendasikan, karena XML menggunakan konsep MVC (*Model-Viewer-Control*) yang memisahkan antara UI dengan kode logika program (*Logic Program*). Dengan begitu, adaptasi program aplikasi dari satu format resolusi layar ke format resolusi yang lain menjadi mudah.

Membuat UI di XML sangat mirip dengan membuat dokumen HTML, contoh sederhana :

```
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    The content of body element
  </body>
</html>
```

1) Hirarki Elemen Layar

Unit dasar dari aplikasi Android adalah *activity*, yaitu sebuah obyek dari kelas `Android.app.Activity`. *Activity* bisa melakukan banyak hal, tetapi dia sendiri tidak akan terlihat secara langsung di layar. Untuk membuat *activity* terlihat di layar adalah

dengan membuatnya sebuah *User Interface*, dalam hal ini dapat menggunakan *Views* dan *ViewGroups*.

a) *Views*

Views adalah obyek dari kelas (*base class*) `Android.view.Views`. Merupakan data struktur yang menyimpan layout dan konten untuk area layar. Obyek *View* menangani ukuran, *layout*, *drawing*, perpindahan fokus, *scrolling*, *key/gesture* untuk area layar.

Kelas *View* tersedia sebagai kelas induk (*base class*) untuk semua *widget*, implementasi dari subkelas yang menggambar elemen interaktif pada layar. *Widget* menangani sendiri bagaimana dia diimplementasikan, sehingga kita bisa memakainya untuk membuat UI secara lebih cepat. *List* dari *widgets* yang tersedia meliputi *TextView*, *EditText*, *Button*, *RadioButton*, *CheckBox*, *ScrollView*, dan lain-lain.

b) *View Groups*

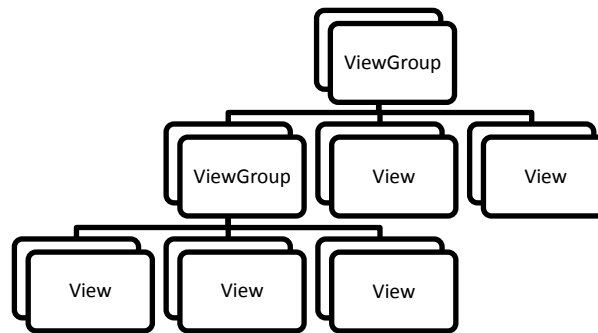
ViewGroups adalah obyek dari kelas `Android.view.ViewGroup`, dan dari namanya mengindikasikan *ViewGroup* adalah tipe khusus dari obyek *View* yang berfungsi untuk mewadahi dan

mengelola *subordinat Views* dan *ViewGroup* yang lain. *ViewGroup* membolehkan programmer untuk menambahkan struktur pada *User Interface* dan membuat elemen layar yang kompleks dan diberikan alamat sebagai satu kesatuan entitas.

Kelas *ViewGroups* tersedia sebagai kelas induk (*base class*) untuk *layout*, implementasi dari subkelas yang menyediakan tipe dasar dari layout layar. Dan layout memberikan cara bagaimana membangun struktur untuk *view*.

2) Hirarki View Android *User Interface*

Pada platform Android, programmer dapat mendefinisikan UI *Activity* dengan menggunakan struktur pohon *view (tree of view)* dan jaringan *ViewGroup* seperti yang terlihat pada gambar. Strukturnya bisa dibuat sederhana atau kompleks, sesuai dengan kebutuhan.



Gambar II.2. Hirarki View Android User Interface
 Sumber : Nicolash Gramlich, andbook! Android
 Programming release 002. <http://andbook.anddev.org/>, 2010.

Agar layar bisa menampilkan tampilan yang sesuai dengan hirarki *view*, *Activity* yang dibuat harus memanggil *method setContentView ()* dan *mempassing* referensi pada obyek yang ingin ditampilkan. Sistem Android kemudian akan menerima referensi itu dan menggunakannya sebagai nilai-nilai tampilan yang akan ditampilkan.

3) Android Manifest

File *AndroidManifest.xml* diperlukan oleh setiap aplikasi Android. Diletakkan pada folder *root* aplikasi, dan mendeskripsikan nilai global paket aplikasi kita. Termasuk komponen yang ada pada aplikasi (*activity*, *service*, dll) yang memang diekspos untuk sistem diluar aplikasi yang membutuhkan. Termasuk juga data pada tiap *Activity*, bagaimana menanganinya dan bagaimana *Activity* tersebut di jalankan (*launched*).

Hal yang penting yang harus diperhatikan dari file `AndroidManifest.xml` adalah *IntentFilters*. *IntentFilters* menjelaskan di mana dan kapan *Activity* dapat dijalankan. Ketika *Activity* atau sistem operasi akan melakukan sesuatu, misalnya membuka halaman web atau membuka layar kontak, itu akan membuat obyek *intent*.

Obyek *intent* ini akan menangani beberapa informasi yang mendeskripsikan apa yang akan dilakukan oleh aplikasi dan data apa yang dibutuhkannya untuk itu, selain juga beberapa informasi lain. Android membandingkan informasi pada obyek *intent* dengan *intentfilter* yang dibuka (*exposed*) oleh setiap aplikasi dan mencari *Activity* yang tepat untuk menangani data atau aksi (*action*) yang dibutuhkan oleh Aplikasi (*Activity*) yang memanggilnya. Jika terdapat lebih dari satu aplikasi yang menangani *intent* tersebut, pengguna (*user*) akan diberi peringatan, melalui pesan atau notifikasi mana yang akan dipilih.

Selain mendeklarasikan *Activity* Aplikasi, *Content Providers*, *Service*, dan *Intent Receivers*, harus juga ditetapkan permission di dalam `AndroidManifest.xml`.

a) Hampir setiap `AndroidManifest.xml` akan memasukkan deklarasi

namespace (`xmlns:Android=http://schemas.Android.com/`

apk/res/ Android) pada elemen pertama. Ini membuat beberapa atribut standar Android tersedia di dalam file, yang akan digunakan untuk mensuply kebanyakan data untuk element di dalam file.

- b) Hampir setiap manifest memasukkan satu tag <application>, yang berisi beberapa tag yang menjelaskan tentang *Applications, IntentReceivers*, dll yang ada pada aplikasi.
- c) Jika programmer ingin agar *Activity* bisa langsung dibuka oleh user, programmer harus membuatnya mendukung MAIN dan LAUNCHER.

Struktur secara detail <tags> dari file AndroidManifest adalah :

- a) <manifest>, merupakan titik *root* utama dari AndroidManifest.xml. Berisi atribut paket (*packge-attribute*) yang menjadi titik dari semua paket pada Activity. Path Activity lain secara relatif akan mengacu pada nilai ini.

```
<manifest
  xmlns:Android=http://schemas.Android.com/apk/res/An
  droid
    package="org.anddev.Android.smsstrecher">
```

- b) <uses-permission>, menjelaskan tentang *security permission* yang harus diberikan agar aplikasi *activity* bisa secara benar beroperasi, misal ketika mengirim sms atau menggunakan

daftar kontak telepon. Permissi ini diberikan kepada user ketika penginstalan aplikasi. Jumlah : 0⁺

```
<uses-permission
```

```
Android:name="Android.permission.RECEIVE_SMS" />
```

- c) `<permission>`, mendeklarasikan *security permission* yang bisa digunakan untuk membatasi aplikasi yang mana yang dapat diakses komponennya atau fiturnya di dalam aplikasi yang dirancang atau di paket aplikasi lain. Jumlah : 0⁺

- d) `<instrumentation>`, mendeklarasikan kode dari komponen instrumen yang tersedia untuk menguji fungsionalitas dari paket aplikasi atau paket aplikasi yang lain. Jumlah : 0⁺

- e) `<application>`, Elemen *root* yang berisi deklarasi dari level aplikasi komponen yang terdapat pada paket aplikasi. Elemen ini juga bisa berisi global dan/atau atribut default dari aplikasi, seperti label, ikon, tema, permissi yang dibutuhkan, dll. Jumlah 0 atau 1.

```
<application Android:icon="@drawable/icon">
```

- f) `<activity>`, merupakan hal mendasar pada sebuah aplikasi untuk berinteraksi dengan user. Setiap *Activity* harus memiliki sebuah `<activity>` pada file manifest, apakah itu untuk diekspos keluar atau diperuntukkan hanya untuk paket aplikasi

itu sendiri. Jika *activity* tidak memiliki tag yang sesuai di dalam manifest, kita tidak bisa menjalankannya. Jumlah : 0⁺

Secara bebas, untuk mendukung *late runtime lookup*, kita bisa menempatkan elemen `<intent-filter>` 1⁺ untuk mendeskripsikan aksi yang didukung *activity*.

g) `<intent-filter>`, mendeklarasikan intent seperti apa yang didukung oleh komponen. Sebagai tambahan kita bisa menempatkan nilai pada elemen ini, atribut bisa diberikan di sini untuk mensupply label, ikon, dan informasi lain untuk aksi (*action*) yang akan dilakukan.

h) `<action>`, tipe aksi (*action-type*) yang didukung oleh komponen. Contoh :

```
<action Android:name="Android.intent.action.MAIN">
```

i) `<category>`, kategori (*category-type*) yang didukung oleh komponen. Contoh :

```
<category
```

```
Android:name="Android.intent.category.LAUNCHER">
```

j) `<data>`, tipe MIME, skema URI, authority URI, atau path URI yang didukung komponen.

k) `<meta-data>`, menambahkan bagian baru metadata kepada *activity*, dimana client akan mendapatkan nilainya kembali.

```
ComponentInfo.metaData.
```

- l) `<receiver>`, sebuah *intentReceiver* mengizinkan aplikasi untuk diberikan informasi mengenai perubahan data atau aksi yang terjadi, bahkan jika sedang berjalan (*running*). Seperti juga dengan tag `<activity>`, secara kita dapat menempatkan lebih dari satu elemen `<intent-filter>` yang didukung oleh receiver atau memasukkan nilai pada `<meta-data>`.

```
<receiver Android:name=".SMSReceiver">
```

- m) `<service>`, merupakan komponen yang dapat berjalan pada *background*. Seperti tag *activity*, secara opsional kita dapat menempatkan satu atau lebih elemen `<intent-filter>` yang didukung oleh *service* atau nilai `<meta-data>`.

- n) `<provider>`, merupakan komponen yang mengelola data (*persistent data*) dan mempublishnya untuk diakses oleh aplikasi lain. Developer secara opsional dapat meletakkan satu atau lebih nilai `<meta-data>`, seperti yang dijelaskan pada `activity <meta-data>`. (Gramlich, 2010 : 19-28).

4) Resources dan R.Java

a) Resources

Resources adalah file eksternal (non-code files) yang digunakan oleh *source code* aplikasi dan dikompile ke dalam aplikasi pada saat build. Android mendukung banyak macam file resources, termasuk file XML, PNG, dan JPEG.

Resource ditempatkan di luar *source code*, dan file XMP di kompile ke dalam binary, format yang bisa cepat di load untuk alasan efisiensi. String dikompres ke dalam format penyimpanan yang lebih efisien.

(1) Daftar Resources

Tipe resource dan tempat di mana mereka ditempatkan :

- a. Layout-files : “/res/layout/”
- b. Images : “/res/drawable/”
- c. Animation : “/res/anim/”
- d. Styles, string dan array : “/res/values/”

Nama file tidak mesti persis dengan :

- a. ‘array.xml’ untuk mendefinisikan array
- b. ‘colors.xml’ untuk mendefinisikan warna #RGB, #ARGB, #RRGGBB, #AARRGGBB
- c. ‘dimens.xml’ untuk mendefinisikan dimensi
- d. ‘strings.xml’ untuk mendefinisikan string

- e. 'styles.xml' untuk mendefinisikan obyek
 - f. Raw file seperti mp3 atau video : `"/res/raw/"`
- (2) Menggunakan Resources di dalam *source code*

Menggunakan resource di dalam *source code* adalah dengan mengetahui ID resource dan tipe dari obyek resource tersebut. Adapun syntax untuk mengacu kepada resource :

```
R.resource_type.resource_name atau
Android.R.resource_type.resource_name
```

Dimana `resource_type` adalah subkelas (*subclass*) dari `R` yang merupakan tipe spesifik dari `resource.resource_name` merupakan atribut dari resource yang didefinisikan pada file XML, atau file name (tanpa ekstensi) untuk resource yang didefinisikan oleh tipe file lain. Setiap tipe dari resource akan ditambahkan pada subkelas `R` yang spesifik, tergantung dari tipe resources.

Resources yang dikompile sendiri oleh aplikasi dapat direfer tanpa nama paket (seperti `R.resource_type.resource_name`).

Android memiliki beberapa standar resource, seperti *style* latar, dan *background button*. Untuk merefer pada kode itu, kita harus memastikan kualifikasinya dengan Android, seperti contoh :

```
Android.R.drawable.button_background.
```

(3) Resource Reference

Nilai yang dimasukkan pada atribut dapat juga direfer pada resource yang lain. Seringkali ini dilakukan pada file *layout* untuk memasukkan *string* sehingga bisa dilocalized dan *image* yang berada pada file lain, walaupun juga bisa dilakukan pada tipe resource lain misalnya warna dan nilai integer.

Sebagai contoh, jika kita memiliki resource warna, kita bisa membuat file layout yang mengeset ukuran dari text color menjadi nilai yang dimiliki oleh resource itu.

```
<EditText
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:textColor="@color/opaque_red"
    Android:text="Hello, World!" />
```

Seperti yang terlihat prefik “@” digunakan untuk memperkenalkan resource reference, teks yang menyertai kemudian adalah nama dari resource di dalam format @[package:]type/name. Pada kasus ini paket tidak dispesifikasikan karena resource direferensi di dalam paket itu sendiri. Untuk mereference sistem resource, dapat dilakukan dengan cara :

```
<EditText
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:textColor="@Android:color/red"
    Android:text="Hello_world!" />
```

atau dalam contoh lain, sebaiknya selalu menggunakan resource reference ketika mensupply string ke dalam file layout sehingga resource tersebut bisa di lokalisir (*localized*)

```
<EditText
    Android:layout_width="fill_parent"
    Android:layout_height="fill_parent"
    Android:textColor="@Android:color/red"
    Android:text="@string/Hello_world!" />
```

b) R.Java

R.Java merupakan file yang digenerate secara otomatis, yang mengindeks semua resource dari proyek aplikasi. Kelas ini bisa digunakan ke dalam *source code* sebagai cara pintas untuk merefer ke *resource* yang di masukkan ke dalam proyek aplikasi. Ini sangat bermanfaat apabila dikaitkan dengan fitur IDE *code-completion* seperti Eclipse karena ini membuat developer secara cepat dan interaktif menemukan spesifik *resource*. Selain itu, developer bisa meningkatkan waktu kompilasi karena resource yang akan dipakai benar-benar ada.

11. Modular Object-Oriented Dynamic Learning Environment (Moodle)

Moodle adalah sebuah nama untuk sebuah program aplikasi yang dapat merubah sebuah media pembelajaran kedalam bentuk web. Aplikasi ini memungkinkan siswa untuk masuk kedalam ruang kelas digital untuk mengakses materi-materi pembelajaran. Moodle merupakan sebuah aplikasi Course Management System (CMS) yang gratis dan digunakan ataupun dimodifikasi oleh siapa saja dengan lisensi secara GNU (General Public License). Aplikasi Moodle dikembangkan pertama kali oleh Martin Dougiamas pada Agustus 2002 dengan Moodle versi 1.0.

Saat ini Moodle bisa dipakai oleh siapa saja secara *Open Source*. Sistem yang dibutuhkan agar aplikasi Moodle ini dapat berjalan dengan baik adalah sebagai berikut :

- Apache Web Server
- PHP
- Database MySQL atau PostgreSQL

Dengan menggunakan Moodle kita dapat membangun sistem dengan konsep *e-learning* (Pembelajaran secara elektronik) ataupun *Distance Learning* (pembelajaran jarak jauh). Dengan konsep ini sistem belajar mengajar akan tidak terbatas ruang dan waktu.

Berikut adalah beberapa management yang terdapat dalam Moodle :

a. Site Management

- Website diatur oleh Admin, yang telah ditetapkan ketika membuat website.
- Tampilan (Themes) diizinkan pada admin untuk memilih warna, jenis huruf, susunan dan lain sebagainya untuk kebutuhan tampilan.
- Bentuk kegiatan yang ada dapat ditambah.
- Source Code yang digunakan ditulis dengan menggunakan PHP.

Mudah untuk dimodifikasi dan sesuai dengan kebutuhan.

b. User management

- Tujuannya ialah untuk mengurangi keterlibatan admin menjadi lebih minimum, ketika menjaga keamanan yang berisiko tinggi.
- Metode Email standar : Pelajar dapat membuat nama pemakai untuk login. Alamat email akan diperiksa melalui konfirmasi.
- Tiap orang disarankan cukup 1 pengguna saja untuk seluruh sever. Dan tiap pengguna dapat mempunyai akses yang berbeda.
- Pengajar mempunyai hak istimewa, sehingga dapat mengubah (memodifikasi) bahan pelajaran.
- Ada “kunci pendaftaran” untuk menjaga akses masuk dari orang yang tidak dikenal
- Semua Pengguna dapat membuat biografi sendiri, serta menambahkan photo.
- Setiap pengguna dapat memilih bahasa yang digunakan. Bahasa Indonesia, Inggris, Jerman, Spanyol, Perancis, dan Portugis dll.

c. Course management

- Pengajar mengendalikan secara penuh untuk mengatur pelajaran, termasuk melarang pengajar yang lain.
- Memilih bentuk/metode pelajaran seperti berdasarkan mingguan, berdasarkan topik atau bentuk diskusi.
- Terdapat Forum, Kuis, Polling, Survey, Tugas, Percakapan dan Pelatihan yang digunakan untuk mendukung proses belajar.

- Semua kelas-kelas untuk forum, Kuis – kuis dan tugas-tugas dapat ditampilkan pada satu halaman (dan dapat didownload sebagai file lembar kerja).
- Bahan pelajaran dapat dipaketkan dengan menggunakan file zip.

12. Unified Modeling Language (UML)

Unified Modeling Language atau lebih sering disebut UML adalah sebuah bahasa untuk menentukan, visualisasi, konstruksi, dan mendokumentasikan *artifact* (bagian dari informasi yang digunakan atau dihasilkan dalam suatu proses pembuatan perangkat lunak. *Artifact* dapat berupa model, deskripsi atau perangkat lunak) dari sistem perangkat lunak, seperti pada pemodelan bisnis dan sistem non perangkat lunak lainnya.

UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam memodelkan sistem yang besar dan kompleks. UML tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Bagian-bagian utama dari UML adalah view, diagram, model element, dan general mechanism.

a. View

View digunakan untuk melihat sistem yang dimodelkan dari beberapa aspek yang berbeda. View bukan melihat grafik, tapi merupakan suatu abstraksi yang berisi sejumlah diagram. Beberapa jenis view dalam UML antara lain: use case view, logical view, component view, concurrency view dan deployment view.

b. Use case view

Mendeskripsikan fungsionalitas sistem yang seharusnya dilakukan sesuai yang diinginkan external actors. Actor yang berinteraksi dengan sistem dapat berupa user atau sistem lainnya.

View ini digambarkan dalam use case diagram dan kadang-kadang dengan activity diagrams. View ini digunakan terutama untuk pelanggan, perancang (designer), pengembang (developer) dan penguji sistem (tester).

c. Logical view

Mendeskripsikan bagaimana fungsionalitas dari sistem, struktur statis (class, object dan relationship) dan kolaborasi dinamis yang terjadi ketika objek mengirim pesan ke objek lain dalam suatu fungsi tertentu.

View ini digambarkan dalam class diagrams untuk struktur statis dan dalam state, sequence, collaboration, dan activity diagram untuk model dinamisnya. View ini digunakan untuk perancang (designer) dan pengembang (developer).

d. Component view

Mendeskripsikan implementasi dan ketergantungan modul. Komponen yang merupakan tipe lainnya dari code module diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administrative lainnya. View ini digambarkan dalam component view dan digunakan untuk pengembang (developer).

e. Concurrency view

Membagi sistem ke dalam proses dan prosesor. View ini digambarkan dalam diagram dinamis (state, sequence, collaboration, dan activity diagrams) dan diagram implementasi (component dan deployment diagrams) serta digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

f. Deployment view

Mendeskripsikan fisik dari sistem seperti komputer dan perangkat (nodes) dan bagaimana hubungannya dengan lainnya. View ini digambarkan dalam deployment diagrams dan digunakan untuk pengembang (developer), pengintegrasi (integrator), dan penguji (tester).

g. Diagram

Diagram berbentuk grafik yang menunjukkan simbol elemen model yang disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu view tertentu dan ketika


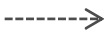
digambarkan biasanya dialokasikan untuk view tertentu. Adapun jenis diagram antara lain :



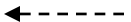


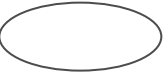

1. Use Case Diagram


Use case adalah abstraksi dari interaksi antara sistem dan actor.

Use case bekerja dengan cara mendeskripsikan tipe interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Use case merupakan konstruksi untuk mendeskripsikan bagaimana sistem akan terlihat di mata user. Sedangkan use case diagram memfasilitasi komunikasi diantara analis dan pengguna serta antara analis dan client. Dibawah ini dapat dilihat simbol-simbol dari use case itu sendiri

Tabel II.1. Simbol Use Case Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		Actor	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
2		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (<i>independent</i>).

3		<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
4		<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
5		<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
6		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).

10		<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan sumber daya komputasi
----	---	-------------	--

2. Class Diagram

Class adalah dekripsi kelompok objek-objek dengan property, perilaku (operasi) dan relasi yang sama. Sehingga dengan adanya class diagram dapat memberikan pandangan global atas sebuah sistem. Hal tersebut tercermin dari class- class yang ada dan relasinya satu dengan yang lainnya. Sebuah sistem biasanya mempunyai beberapa class diagram. Class diagram sangat membantu dalam visualisasi struktur kelas dari suatu system.

3. Component Diagram

Component software merupakan bagian fisik dari sebuah sistem, karena menetap di komputer tidak berada di benak para analis. Komponen merupakan implementasi software dari sebuah atau lebih class. Komponen dapat berupa source code, komponen biner, atau executable component. Sebuah komponen berisi informasi tentang logic class atau class yang diimplementasikan sehingga membuat pemetaan dari logical view ke komponen view. Sehingga komponen diagram merepresentasikan dunia riil yaitu komponen software yang mengandung komponen, interface dan relationship.

4. Deployment Diagram

Menggambarkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware, menunjukkan hubungan komputer dengan perangkat (nodes) satu sama lain dan jenis hubungannya. Di dalam nodes, executeable component dan objek yang dialokasikan untuk memperlihatkan unit perangkat lunak yang dieksekusi oleh node tertentu dan ketergantungan komponen.

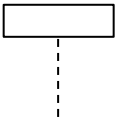


5. State Diagram

Menggambarkan semua state (kondisi) yang dimiliki oleh suatu object dari suatu class dan keadaan yang menyebabkan state berubah. Kejadian dapat berupa object lain yang mengirim pesan. State class tidak digambarkan untuk semua class, hanya yang mempunyai sejumlah state yang terdefinisi dengan baik dan kondisi class berubah oleh state yang berbeda.

6. Sequence Diagram

Sequence Diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Pada halaman disebelah dapat dilihat simbol-simbol dari sequence diagram :

Tabel II.2. Simbol Sequence Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		<i>Objek</i>	Spesifikasi dari field-field atau sub menu yang dapat diakses oleh <i>actor</i>

7. Collaboration Diagram








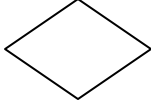
Menggambarkan kolaborasi dinamis seperti sequence diagrams. Dalam menunjukkan pertukaran pesan, collaboration diagrams menggambarkan objek dan hubungannya (mengacu ke konteks). Jika penekannya pada waktu atau urutan gunakan sequence diagrams, tapi jika penekanannya pada konteks gunakan collaboration diagram.

8. Activity Diagram

Menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.

Berikut adalah simbol-simbol yang sering digunakan pada tahap perancangan *activity diagram* :

Tabel II.3. Simbol Activity Diagram

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi.
3		<i>Initial Node</i>	Titik awal
4		<i>Activity Final Node</i>	Titik akhir
5		<i>Fork Node</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran
6		<i>Join</i>	Penggabungan beberapa state
7		<i>Rake</i>	Percabangan beberapa state
8		<i>Decision</i>	Digunakan untuk penyeleksian kondisi didalam <i>activity diagram</i>

BAB III

METODE PENELITIAN

A. Lokasi Penelitian

Adapun tempat yang digunakan untuk melakukan penelitian adalah SMP Negeri 3 Pallangga yang beralamat di Jl. Benteng Somba Opu, Sungguminasa Gowa

B. Jenis Penelitian

Jenis penelitian yang digunakan dalam skripsi ini adalah jenis penelitian deskriptif kualitatif. Penelitian deskriptif kualitatif adalah penelitian yang menggambarkan fakta, keadaan, variabel dan fenomena-fenomena objek ketika penelitian berlangsung sesuai dengan apa adanya. Dalam hal ini, yang akan menjadi prioritas utama adalah modul materi yang akan ditampilkan, serta file-file yang dapat didownload.

C. Metode Pengumpulan Data

Metode pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut :

1. Penelitian Kepustakaan (*Library Research*)

Yaitu penelitian yang dilakukan dengan cara mengumpulkan data yang terdapat dalam buku-buku literature, peraturan perundang-undangan, majalah, surat kabar, hasil seminar dan sumber lain yang terkait dengan perancangan aplikasi pembelajaran.

2. Penelitian Lapangan (*Field Research*)

Yaitu melakukan penelitian langsung ke lokasi penelitian untuk memperoleh data yang valid mengenai objek yang diteliti. Dalam hal ini tentang bahan pembelajaran mata pelajaran Teknologi Informasi dan Komunikasi. Dengan cara melakukan wawancara langsung kepada sumbernya dan melakukan pengamatan secara langsung ke objek penelitian ataupun dengan kurikulum yang berlaku.

D. Hardware dan Software

1. Hardware yang digunakan dalam penelitian ini adalah :

a. Handset Android

Dalam penelitian ini handset yang digunakan adalah Advan T1C Processor Qualcomm MSM7227 ARM11 800 MHz, RAM 512 MB dengan Sistem Operasi Android versi 2.3 (Gingerbread).

b. Laptop/Komputer

Dalam penelitian ini laptop yang digunakan laptop Anote M550SE dengan Processor Intel T2130 (1.86 Ghz, RAM 2 GB). Dengan Sistem Operasi Microsoft Windows Seven, 32 bit.

2. Software yang digunakan dalam penelitian ini adalah :

a. Eclipse dan Android SDK

Dalam penelitian ini, menggunakan aplikasi Eclipse Galileo dan Android SDK dalam membangun aplikasi.

b. JDK

Dalam penelitian ini, software development kit yang digunakan dalam pemrograman bahasa java. JDK yang digunakan ada versi 622-windows i586.

E. System Requirement

Aplikasi yang dirancang adalah aplikasi *client-server*, dan kebutuhan sistem minimal dari aplikasi ini adalah :

1. Perangkat klien
Sistem Operasi Android versi 2.3 (Gingerbread).
2. Perangkat server
Sistem Operasi Microsoft Windows Seven.

F. Metode Pengujian Perangkat Lunak

Pengujian dimaksudkan untuk mengetahui apakah perangkat lunak yang dibuat telah memenuhi tujuan dan menunjukkan bahwa aplikasi yang dirancang memiliki jaminan kualitas yang baik. Secara umum teknik pengujian dilakukan dengan teknik *white box* dan *black box*. Dalam penelitian ini metode yang digunakan adalah metode pengujian *black box*, dimana teknik ini merupakan pengujian analisa system memperoleh kumpulan kondisi input yang akan mengerjakan seluruh keperluan fungsional program. ”*Black box* merupakan salah satu metode pengujian

yang memecah atau membagi domain input dari program ke dalam kelas-kelas data sehingga *test case* dapat diperoleh (Roger S Pressman : 2002)

Pada Black-Box Testing, cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari unit itu sesuai dengan proses bisnis yang diinginkan (Al-Fatta, 2007 : 172).



BAB IV

ANALISIS DAN DESAIN SISTEM

A. Analisis Sistem

Menjelaskan langkah dan tujuan analisis terhadap objek penelitian yang akan diteliti.

1. Analisis Kebutuhan Sistem

a. Kebutuhan Perangkat Keras

- 1) PC Server : Processor Intel T2130 (1.86 Ghz, RAM 2 GB).
- 2) Perangkat mobile (client) : Smartphone yang memiliki system operasi Android versi 2.3 (Gingerbread).

b. Kebutuhan Perangkat Lunak

Software JRE (*Java Runtime Environment*) dan Java SDK (*Standart Development Kit*), JRE digunakan untuk menjalankan aplikasi yang berbasis java, pembangunan sistem berupa aplikasi yang nantinya akan digunakan untuk berhubungan dengan internet.

Eclipse Galileo dan Android SDK adalah aplikasi yang dibutuhkan dalam pembuatan aplikasi berbasis Android. Eclipse merupakan IDE untuk pembuatan aplikasi dan Android SDK merupakan library yang harus dimiliki untuk menjalankan program.

c. Kebutuhan Informasi

Informasi yang akan dihasilkan dari aplikasi ini dapat mempermudah siswa untuk mengakses materi-materi pembelajaran khususnya mata pelajaran Teknologi Informasi dan Komunikasi melalui handset Android.

d. Kebutuhan Pengguna

Dalam penelitian ini yang berperan dalam pembangunan aplikasi adalah *student*, *teacher*, dan *administrator*. Hal – hal yang menjadi aspek kebutuhan *user*, *teacher*, dan *administrator* yaitu :

- 1) Kebutuhan *student* ialah aspek kemudahan dalam mengakses materi-materi pembelajaran Teknologi Informasi dan Komunikasi dimanapun dan kapanpun selama berada dalam jangkauan internet.
- 2) Kebutuhan *teacher* ialah aspek memberikan materi-materi pembelajaran kepada user yang telah mengunduh aplikasi ini.
- 3) Kebutuhan *administrator* ialah aspek *mobilitas* yang melakukan koneksi antara *teacher* dan *student*, serta memberikan aturan-aturan yang berlaku untuk *lalulintas* dari aplikasi pembelajaran ini.

2. Analisis Kelayakan Sistem

a. Kelayakan Teknologi

Teknologi yang akan digunakan pada aplikasi berbasis mobile untuk mengakses materi-materi pembelajaran yaitu teknologi berbasis pemrograman mobile. Teknologi ini sudah umum digunakan dalam bidang Informasi dan Teknologi dan telah dikembangkan ke sistem yang lebih kompleks. Dewasa ini, Teknologi berbasis pemrograman mobile ini mudah ditemukan dalam kehidupan sehari-hari. Karena perangkat mobile seperti *handphone* dan *smartphone* adalah perangkat yang paling dekat dengan manusia.

b. Kelayakan Hukum

Pembuatan aplikasi berbasis android untuk mengakses materi-materi pembelajaran untuk mata pelajaran Teknologi Informasi dan Komunikasi tidak akan melanggar hukum atau peraturan – peraturan yang berlaku baik yang telah ditetapkan oleh pemerintah.

c. Kelayakan Operasional

Pada kelayakan Operasional dibagi menjadi 2 aspek yaitu:

1. Aspek teknis

Dengan dibangunnya aplikasi yang mengkombinasikan antara *e-learning* dan *m-learning* ini, user (siswa) akan diberi kemudahan

dalam mengakses materi-materi pembelajaran khususnya untuk mata pelajaran Teknologi Informasi dan Komunikasi.

2. Aspek psikologis

Sistem ini mudah diterapkan, karena tidak membutuhkan pelatihan yang rumit.

B. Deskripsi Umum Perangkat Lunak

Desain adalah langkah pertama dalam fase pengembangan bagi setiap produk atau sistem yang direkayasa. Desain dapat didefinisikan sebagai “proses aplikasi berbagai teknik dan prinsip bagi tujuan pendefinisian suatu perangkat, suatu proses atau sistem dalam detail yang memadai untuk memungkinkan realisasi fisiknya” .

Tujuan desainer adalah untuk menghasilkan suatu model atau representasi dari entitas yang kemudian akan dibangun. Proses dimana model dikembangkan menggabungkan intuisi dan penilaian berdasarkan pengalaman dalam membangun entitas yang sama, serangkaian prinsip dan heuristik yang menuntun cara dimana model disusun, serangkaian kriteria yang memungkinkan penilaian terhadap kualitas, dan suatu proses iterasi yang membawa kepada suatu representasi desain akhir (Pressman, 2002 : 399).

Output utama dari tahapan desain perangkat lunak adalah spesifikasi desain. Spesifikasi ini meliputi spesifikasi desain umum yang akan disampaikan kepada stakeholder sistem dan spesifikasi desain rinci yang akan digunakan pada tahap implementasi. Spesifikasi desain umum hanya berisi gambaran umum agar

stackholder sistem mengerti akan seperti apa perangkat lunak yang akan dibangun. Spesifikasi desain rinci atau kadang disebut desain arsitektur rinci perangkat lunak diperlukan untuk merancang sistem sehingga memiliki konstruksi yang baik, proses pengolahan data yang tepat dan akurat, bernilai, memiliki aspek *user friendly* dan memiliki dasar-dasar untuk pengembangan selanjutnya.

Desain perangkat lunak berada pada inti teknik dari proses rekayasa perangkat lunak dan diaplikasikan tanpa memperhatikan model proses perangkat lunak yang digunakan. Begitu persyaratan perangkat lunak telah mulai dianalisis dan ditentukan, maka desain perangkat lunak menjadi yang pertama dari tiga aktifitas teknik (desain, pembuatan kode, dan pengujian) yang diperlukan untuk membangun dan menguji perangkat lunak. Masing-masing aktifitas memindahkan informasi dengan suatu cara yang terutama menghasilkan perangkat lunak komputer yang tervalidasi.

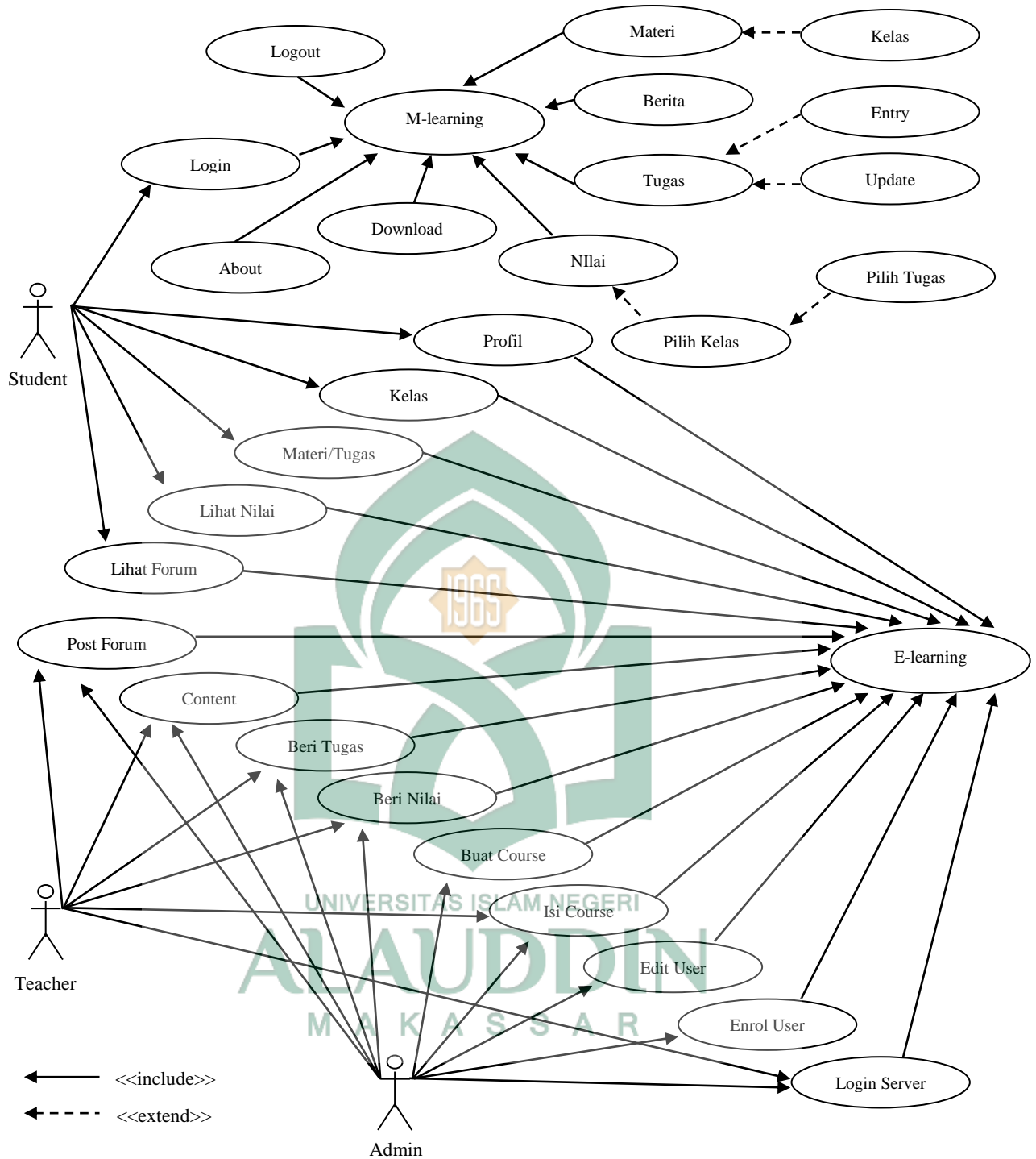
C. Perancangan Sistem

1. Proses

Tahap perancangan dimulai dengan pembuatan UML atau Unified Modelling Language untuk mengetahui perilaku bagaimana sistem bereaksi. Perilaku ini merupakan aktifitas sistem yang bisa dilihat dari luar dan bisa diuji.

Untuk memperjelas UML dari *moodle* dan *m-learning* itu sendiri, dapat dilihat pada Use Case, Activity Diagram serta Sequence Diagramnya, berikut adalah gambar Use Case dari *m-learning* yang dirancang.

a. Use Case diagram aplikasi *m-learning* yang dirancang



Gambar IV.1. Use Case Diagram

b. Activity Diagram

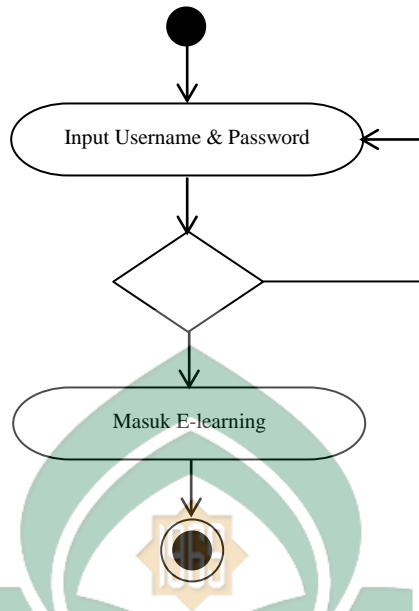
Activity diagram merupakan bentuk *flow diagram* yang memodelkan alur kerja sebuah proses dan urutan aktivitas sebuah proses. Diagram ini mirip dengan *flowchart* karena dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya. *Activity diagram* juga berguna untuk menggambarkan perilaku paralel atau bagaimana perilaku dalam berbagai *use case* berinteraksi. Berikut adalah *activity diagram* secara keseluruhan proses yang terjadi di dalam system aplikasi mobile learning meliputi User (siswa), teacher dan admin.

Sebelum siswa dapat mengakses *m-learning*, diharuskan untuk memasukkan username dan password yang sebelumnya telah diinput oleh *admin*.



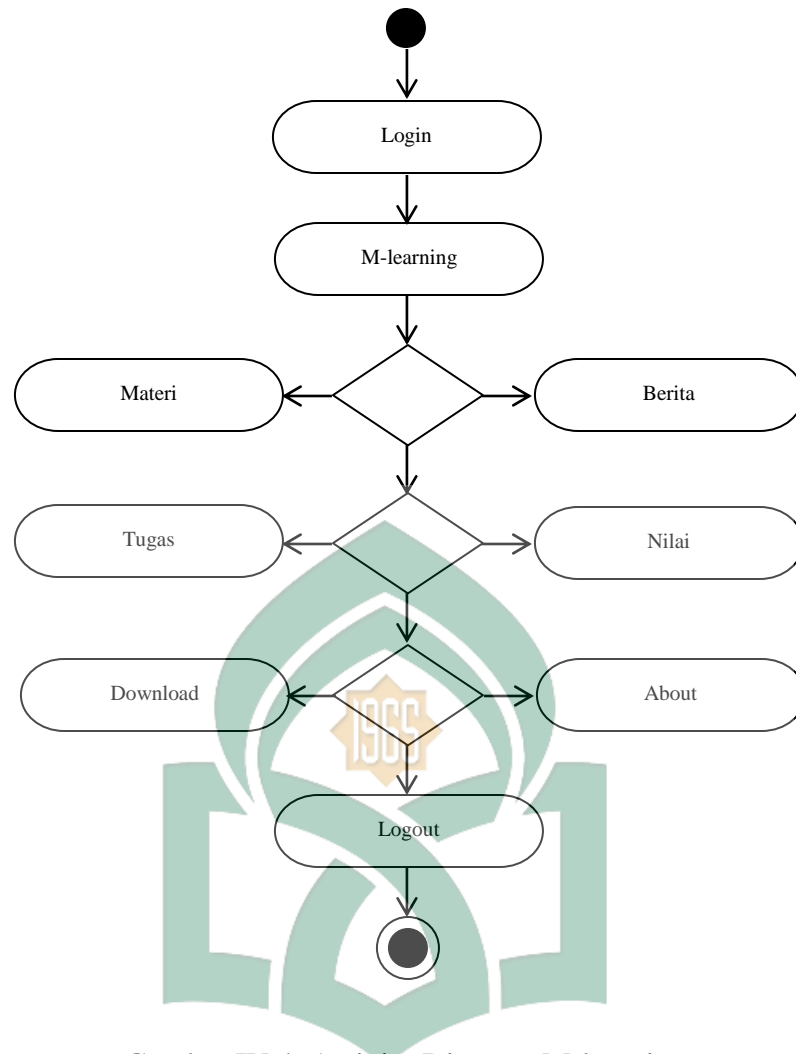
Gambar IV.2. Activity Diagram User

Untuk *e-learning* (moodle) dapat diakses oleh *student*, *teacher* dan *admin* dengan *account* yang sebelumnya telah didaftar oleh *admin*. Sedangkan untuk mengakses *server* hanya dapat dilakukan oleh *teacher* dan *admin*.



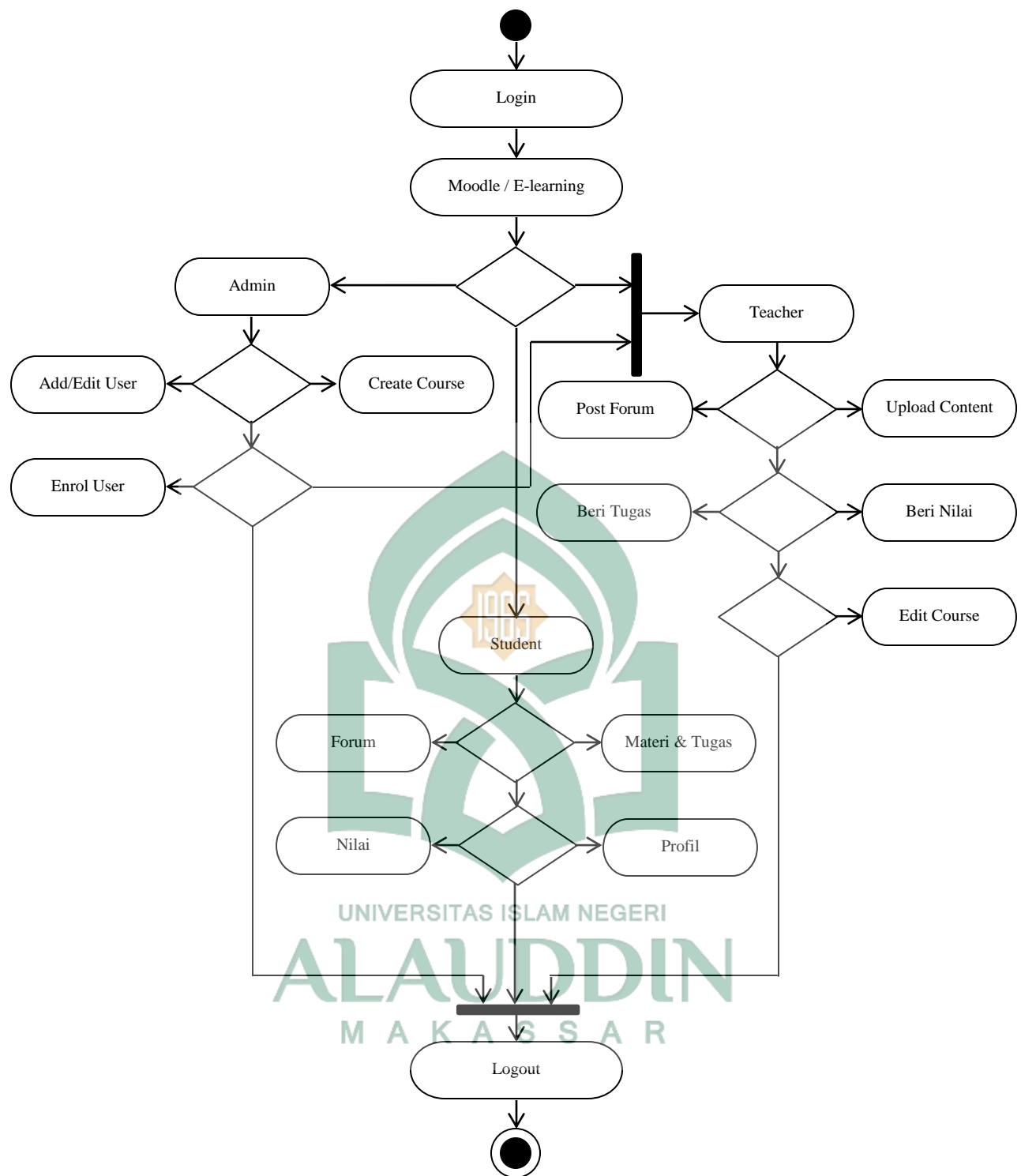
Gambar IV.3. Activity Diagram Teacher dan Admin

Adapun *activity diagram* untuk user yang sedang mengakses *m-learning* dapat memilih beberapa item yang disediakan dalam aplikasi ini, terdiri dari materi, berita, tugas, nilai, download, about dan logout. Untuk item materi, user akan diperlihatkan beberapa pilihan kelas terlebih dahulu atau dengan kata lain pilihan kelas yang telah di-*enrol* oleh admin menurut database siswa (user) yang ada. Sedangkan untuk item lainnya akan mengikuti pilihan kelas yang telah di-*enrol* terlebih dahulu.



Gambar IV.4. Activity Diagram M-learning

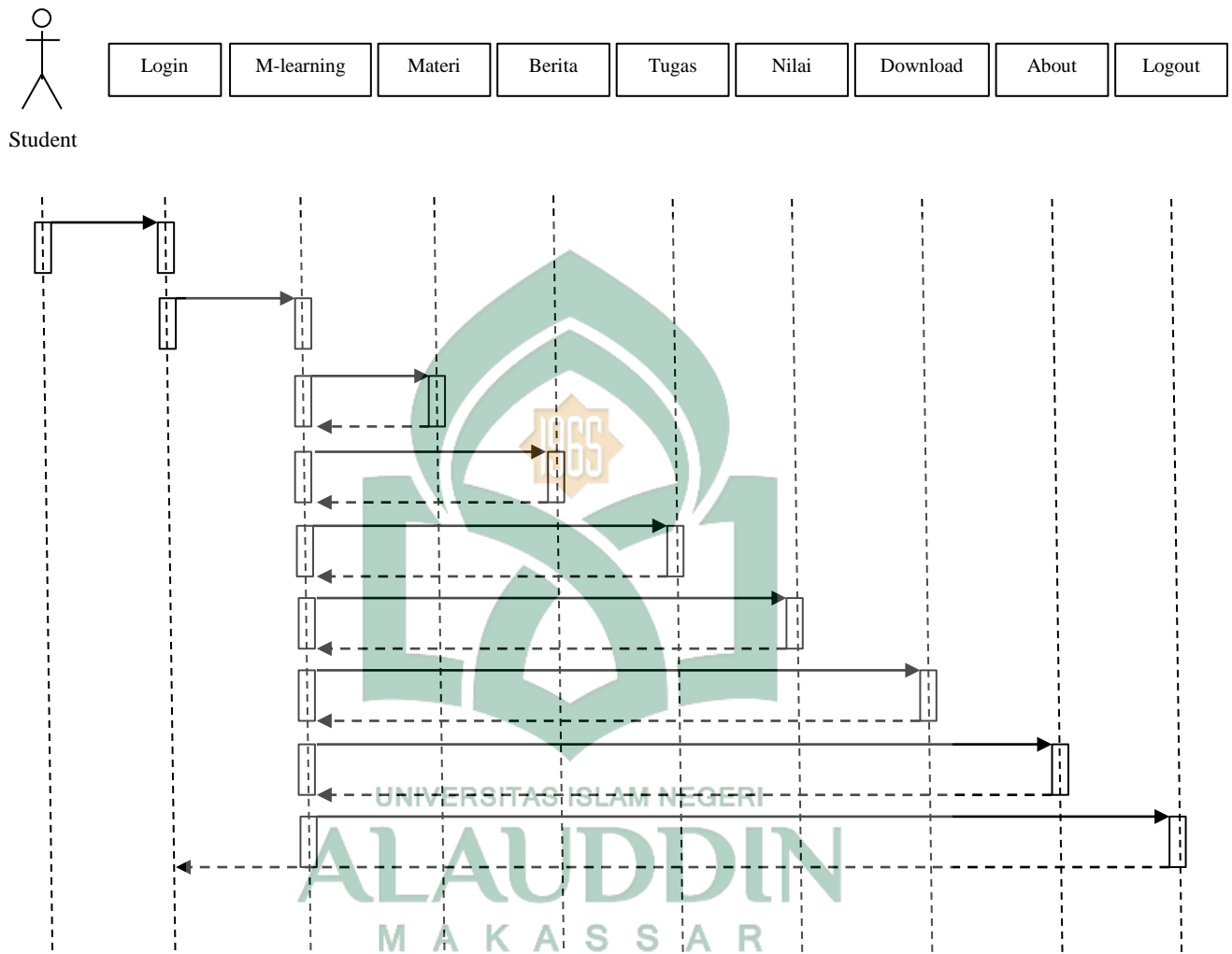
Activity diagram berikutnya adalah activity diagram untuk *student*, *teacher* dan *admin* dalam pengaksesan *e-learning* (moodle) yang akan dipergunakan untuk *server* dalam keperluan teknis dari aplikasi yang dirancang dan sebagai bahan materi untuk *m-learning*.



Gambar IV.5. Activity Diagram E-learning (moodle)

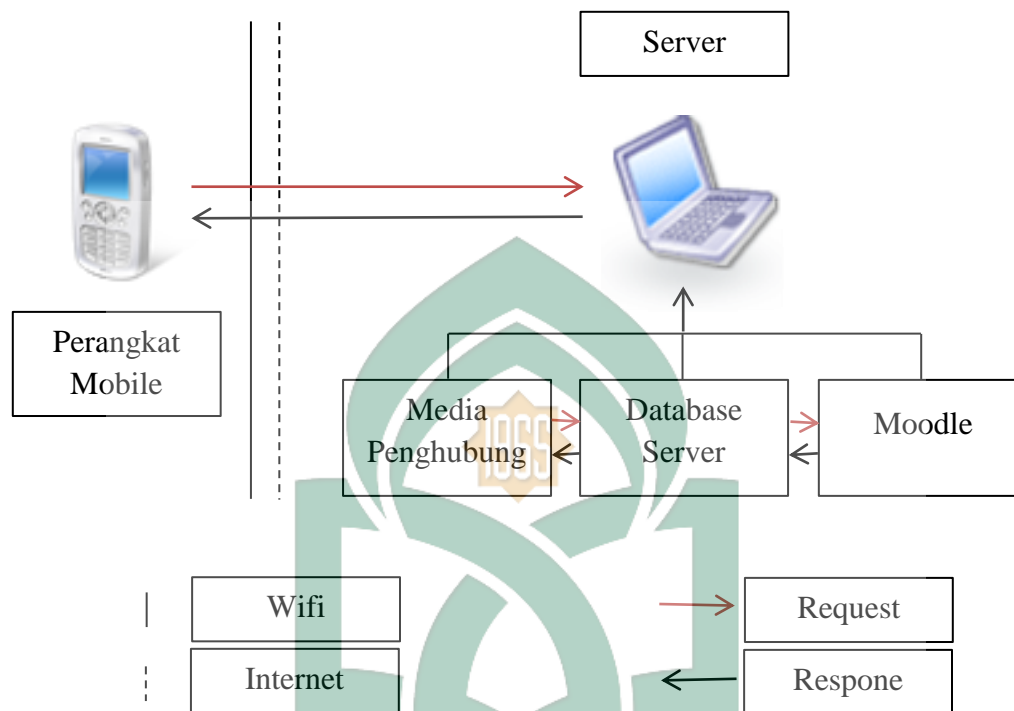
c. Sequence Diagram

Sequence diagram menjelaskan interaksi obyek yang tersusun dalam suatu urutan waktu. Dalam *sequence diagram* diperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam *use case*.



Gambar IV.6. Sequence Diagram

Aplikasi pembelajaran *m-learning* yang telah dibuat dapat diakses melalui perangkat mobile, ketika perangkat mobile android terhubung dengan server menggunakan fasilitas wifi yang dipancarkan oleh server atau dengan menggunakan akses internet maka materi-materi yang terdapat pada aplikasi pembelajaran (moodle) dapat diakses.

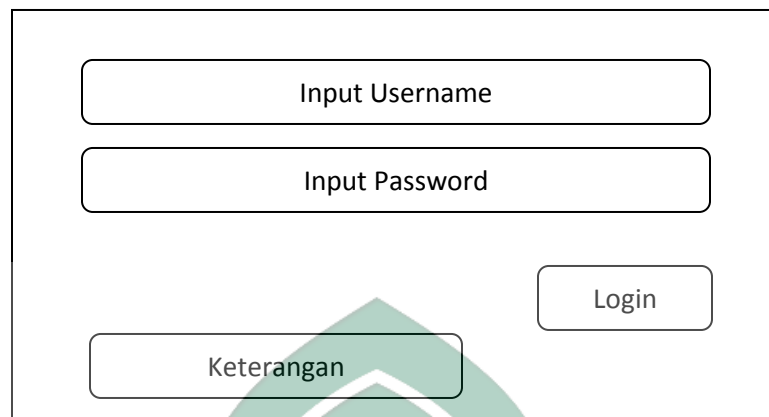


Gambar IV.7. Diagram Rancangan Sistem

2. Perancangan Antarmuka / *Interface*

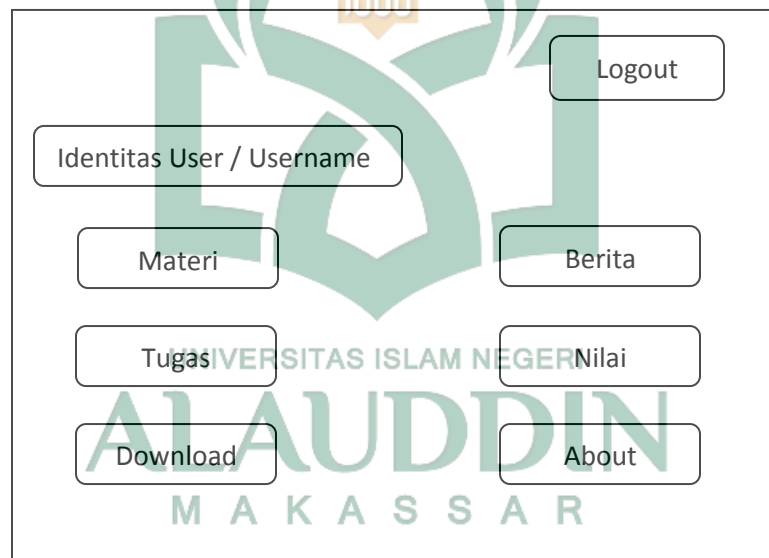
Perancangan antarmuka mengandung penjelasan tentang rancangan antar muka dari aplikasi perancangan form-form yang ada dalam aplikasi.

Adapun rancangan *interface* pada aplikasi pembelajaran interaktif ini adalah :



The login window design features a central area with four rounded rectangular buttons. At the top, there is an 'Input Username' button, followed by an 'Input Password' button. To the right of the password input is a 'Login' button. Below the password input is a 'Keterangan' (Description) button. A large, faint watermark of the Universitas Islam Negeri Alauddin Makassar logo is visible in the background.

Gambar IV.8. Rancangan Jendela Login



The main page design features a grid of buttons. At the top right is a 'Logout' button. Below it, on the left, is a button labeled 'Identitas User / Username'. In the center, there are four buttons arranged in a 2x2 grid: 'Materi' (top left), 'Berita' (top right), 'Tugas' (bottom left), and 'Nilai' (bottom right). At the bottom, there are two buttons: 'Download' on the left and 'About' on the right. A large, faint watermark of the Universitas Islam Negeri Alauddin Makassar logo is visible in the background.

Gambar IV.9. Rancangan Halaman Utama

Untuk rancangan antarmuka / *interface* dari aplikasi pembelajaran interaktif ini digunakan bahasa pemrograman Java karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, seperti Symbian dan Android.

Aplikasi pembelajaran interaktif ini adalah aplikasi *client-server*. Sedangkan untuk mensinkronisasikan antara *client* yang menggunakan *m-learning* dan *server* yang menggunakan *e-learning* atau moodle, maka dibutuhkan *connected file* yang dibuat dengan menggunakan PHP yang akan menghubungkan antara aplikasi *mobile* dan aplikasi moodle yang ada pada *server*.



BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

A. Implementasi Perangkat Lunak

Aplikasi *mobile-learning (m-learning)* ini diimplementasikan untuk sistem operasi Android yang berjalan pada perangkat telepon genggam. Aplikasi ini dirancang menggunakan perangkat lunak atau *software* aplikasi Eclipse Galileo dan *software development kit* yang digunakan dalam pemrograman bahasa java. JDK yang digunakan adalah versi 622-windows i586.

B. Pengujian Sistem

Pengujian sistem aplikasi difokuskan aplikasi *e-learning* dan *m-learning* itu sendiri dengan menggunakan *account student*. Untuk pengujian sistem aplikasi *m-learning*, perangkat mobile harus memiliki koneksi dengan internet atau jaringan Wi-Fi ataupun *packet data* dari kartu provider dalam *handphone*. Adapun pengujian sistem yang telah dilakukan adalah :

a. Jendela Login Screen E-learning

Saat aplikasi *e-learning* pertama kali dijalankan maka akan ditampilkan *Login Screen*. Pada tampilan ini, user diharuskan memasukkan username dan password yang sebelumnya telah diregistrasi oleh *administrator*.

YOU ARE NOT LOGGED IN. (LOGIN)

E-Learning TIK
Tempat yang tepat menggapai ilmu

CONTACT US
inho_joker@yahoo.com
Administrator

TIK ► LOGIN TO THE SITE English (en)

RETURNING TO THIS WEB SITE?
Login here using your username and password
(Cookies must be enabled in your browser) ⓘ

USERNAME PASSWORD Some courses may allow guest access

Forgotten your username or password?

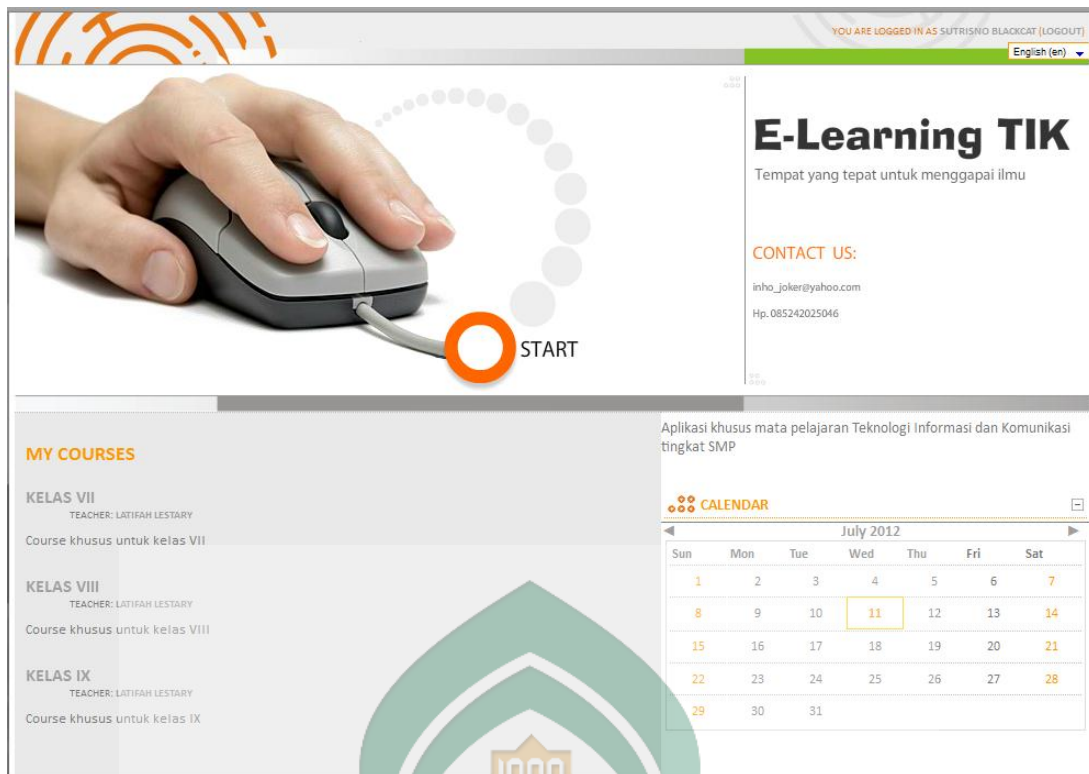
[Home](#)

YOU ARE NOT LOGGED IN. (LOGIN)

Gambar V.1. Jendela Login Screen E-learning

b. Jendela Halaman Utama E-learning

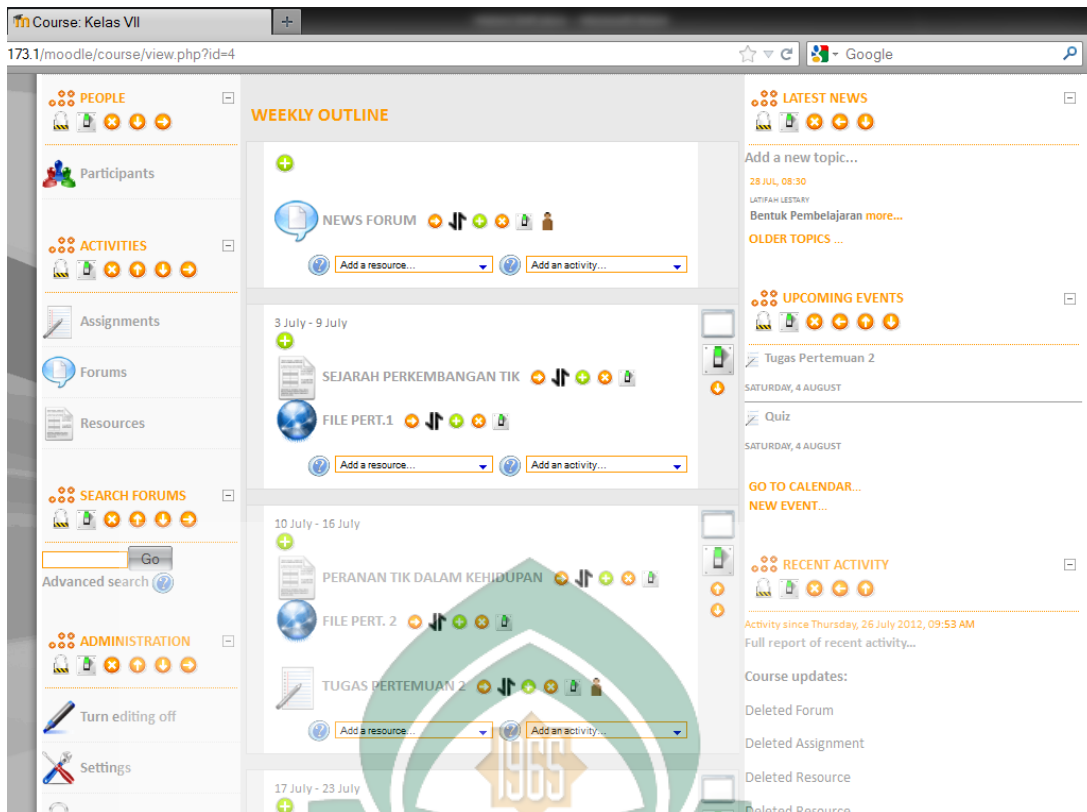
Setelah *student* melakukan *login* pada *e-learning*, maka akan tampil halaman utama yang berisi tentang *course* yang diambil oleh *student*. Course yang dimaksud adalah tingkatan kelas, pada halaman utama dapat kita lihat ada kelas VII, kelas VIII dan kelas IX. *Student* hanya memilih kelas yang diikuti, maka setelah itu akan muncul halaman materi sesuai dengan kelas yang diikuti oleh *student*.



Gambar V.2. Jendela Halaman Utama E-learning

c. Jendela Halaman Materi E-learning

Setelah memilih kelas yang diikuti, maka akan tampil halaman materi yang terdiri dari beberapa pertemuan. Dan dari beberapa pertemuan itu sendiri, terdapat pula pertemuan yang dikhususkan untuk *quiz* dan pertemuan untuk *final*.



Gambar V.3. Jendela Halaman Materi E-learning

d. Jendela Halaman Login M-learning

Saat aplikasi *m-learning* pertama kali dijalankan maka akan ditampilkan *LogIn Screen*. Pada tampilan ini, user diharuskan memasukkan username dan password yang sebelumnya telah diregistrasi pada *server* oleh *administrator*.



Gambar V.4. Jendela Login Screen M-learning

e. Jendela Halaman Utama M-learning

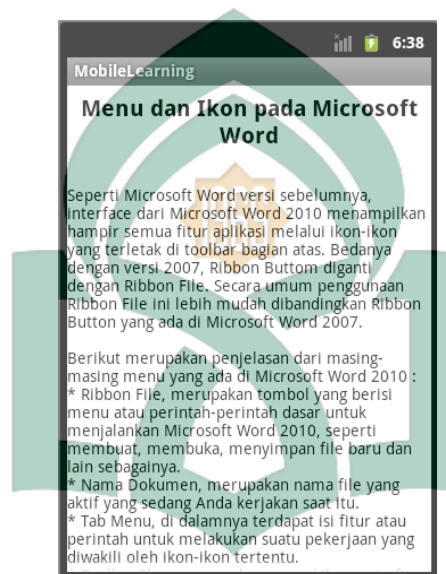
Halaman utama atau welcome screen adalah jendela yang berisi tentang beberapa fasilitas yang dapat diakses oleh user, yaitu dengan mengklik *button image* berupa materi, tugas, berita dan lain sebagainya.



Gambar V.5. Jendela Halaman Utama

f. Jendela Materi

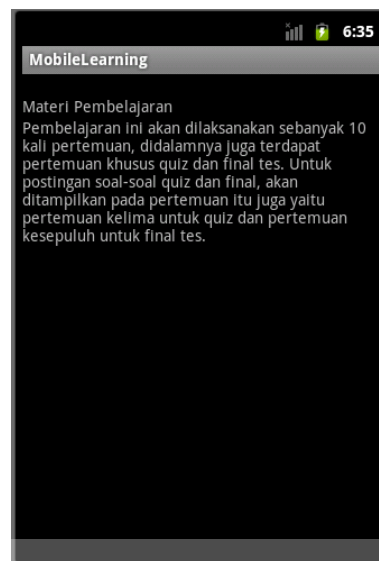
User akan mendapatkan materi-materi pembelajaran pada jendela materi, karena setiap materi yang diupload oleh *teacher* dapat dilihat pada jendela ini. Pada jendela ini terdapat pilihan kelas yang akan disesuaikan dengan kelas dari user itu sendiri. Jika user masih berada pada tingkat pertama, maka yang akan muncul hanya *course* untuk kelas VII dan sebaliknya, jika user telah ada pada tingkat akhir, maka *course* yang akan muncul untuk kelas VII, kelas VIII dan kelas IX.



Gambar V.6. Jendela Materi

g. Jendela Berita

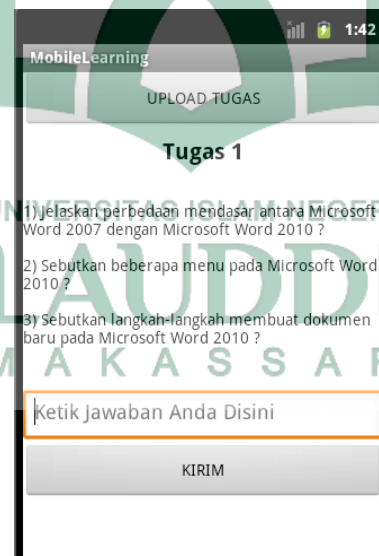
Pada halaman ini, user akan dapat melihat postingan forum atau berita yang diposting oleh guru dari mata kuliah yang bersangkutan. Isi dari content berita ini dapat berupa pemberitahuan ataupun jadwal quiz dan final. Berikut adalah tampilan dari halaman atau jendela berita.



Gambar V.7. Jendela Berita

h. Jendela Tugas

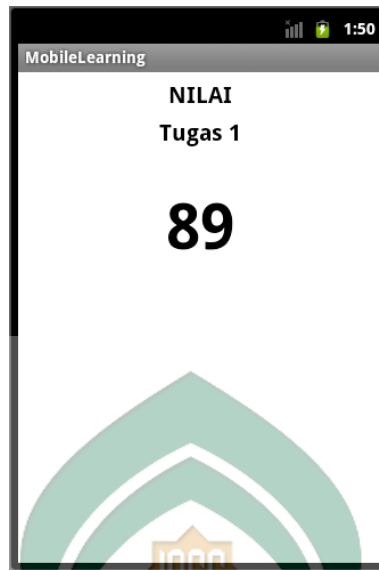
Menampilkan beberapa tugas adalah fungsi dari item tugas itu sendiri. Pada jendela tugas, user akan diarahkan untuk menjawab ataupun mengerjakan tugas-tugas yang telah diupload oleh *teacher*.



Gambar V.8. Jendela Tugas

i. Jendela Nilai

Halaman ini menampilkan penilaian guru terhadap tugas-tugas yang telah dikerjakan oleh siswa (user).



Gambar V.9. Jendela Nilai

j. Jendela Download

Materi-materi yang berhubungan dengan topik yang dibahas, file lengkapnya dapat di unduh pada item download itu sendiri. File yang dapat di download antara lain file document, musik dan video. File-file itu sendiri dapat dipaketkan menjadi satu paket dalam bentuk .zip ataupun .rar untuk setiap pertemuan.



Gambar V.10. Jendela Download

k. Jendela About

Halaman yang berisikan tentang tujuan dari aplikasi pembelajaran interaktif dan info lainnya.



Gambar V.11. Jendela About

BAB VI

KESIMPULAN DAN SARAN

A. Kesimpulan

Dari pembahasan sebelumnya dapat disimpulkan :

1. Aplikasi *m-learning* ini adalah aplikasi *client-server*. Dimana *e-learning* dijadikan sebagai *server* dan merupakan perangkat komputer/laptop dan *client* adalah perangkat *mobile*.
2. Dengan aplikasi ini, proses belajar mengajar khususnya untuk mata pelajaran Teknologi Informasi dan Komunikasi dapat dilakukan dimana saja dengan menggunakan jaringan *internet* atau jaringan *wifi* yang dipancarkan oleh *server*.
3. Aplikasi *e-learning* berjalan selaras dengan aplikasi *m-learning*. Khusus untuk aplikasi *m-learning* dapat diakses pada perangkat *mobile* berbasis Android.

B. Saran

Berdasarkan hasil penelitian dan kesimpulan, maka disarankan :

1. Aplikasi ini dapat digunakan semaksimal mungkin dalam proses belajar mengajar.
2. Aplikasi ini dapat dikembangkan dengan menambahkan *fitur-fitur* yang belum ada dan dikembangkan agar dapat berjalan di semua *platform* sistem operasi *mobile*.

DAFTAR PUSTAKA

- Adri, Muhammad, *Pemanfaatan Multimedia Pembelajaran*, (Pusat Teknologi Informasi dan Komunikasi Depdiknas: 2008).
- Agung Hendranto, *Kiat Jitu Menjadi EDP Andal*, Penerbit Elex Media Komputindo 2006.
- Agus Purwanto, *E-learning Teori Awal Multimedia*, PT. Elex Multimedia Komputindo, Jakarta, 2008.
- Amiruddin Kasim, *Multimedia Pembelajaran Teknologi Informasi dan Komunikasi Sekolah Menengah Atas*, Skripsi, Makassar, 2009.
- Anararki, Firouz, *A Flash-Based Mobile Learning System for Learning English as a Second Language*, International Conference on Computer Engineering Technology : IEEE Computer Society, 2009.
- Dulleng Fakhruddin, dkk, *Pendidikan Agama Islam*, Makassar, 2005.
- Hanif Al Fatta, *Analisis dan Perancangan Sistem Informasi : untuk Keunggulan Bersaing Perusahaan dan Organisasi Modern*, Andi Offset, Yogyakarta, 2007.
- Ir. Yuniar Supardi, *Semua Bisa Menjadi Programmer Android Basic*, PT. Elex Media Komputindo, Jakarta, 2011.
- Iwan Binanto, *Multimedia Digital Dasar Teori + Pengembangannya*, Penerbit Andi, Yogyakarta, 2010.
- Jack Febrian, *Kamus Komputer dan Teknologi Informasi*, Penerbit Informatika, Bandung, 2010.
- Mark L. Murphy, *Beginning Android 2*. United States of America: Aapress. 2010.

Madcoms, *Menghubungkan Data antar Program Microsoft Office 2010*, Penertbit Andi, Yogyakarta, 2011.

Mulyadi, *Android App Inventor*, Multimedia Center Publishing, Yogyakarta, 2011.

Nazruddin Safaat H, *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Penerbit Informatika, Bandung, 2011.

Nazruddin Safaat H, *Android Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android, Edisi Revisi* Penerbit Informatika, Bandung, 2012.

Nicolash Gramlich, *andbook! Android Programming*, rilis 2, <http://andbook.anddev.org/>

Prof. Dr. S. Nasution, M.A., *Berbagai Pendekatan dalam Proses Belajar dan Mengajar*, Bumi Aksara, Jakarta, 2010.

Reto Meier, *Professional Android 2 Application Development*, Indiana : Wiley Publishing, 2010.

Roger S Pressman, *Software Engineering : A Practitioner's Approach* Diterjemahkan oleh LN Harnaningrum dengan judul *Rekayasa Perangkat Lunak : Pendekatan Praktisi (Buku I)*. Andi, Yogyakarta, 2002.

Salameh, Osama, *A Multimedia Offline Cell Phone System For English Language Learning*. *International Arab Journal of e-Technology*, Vol 2, No 1, Januari 2011

Supardi, Yuniar, *Belajar Semua Edisi Java2 Untuk Segala Tingkat*, PT. Elex Media Komputindo, Jakarta, 2009.

Wahana Komputer, *Microsoft Office 2010*, PT. Elex Media Komputindo, Jakarta, 2011.

Wahana Komputer, *Mudah Membuat Video Tutorial dengan Camtasia Studio 7*, PT. Elex Media Komputindo, Jakarta, 2011.

Widodo, Chomsin S, *Panduan Menyusun Bahan Ajar Berbasis Kompetensi*, Jakarta, 2008.

Wiharsono Kurniawan, *Jaringan Komputer*, Penerbit Andi, Yogyakarta, 2007.

Yayasan Penyelenggara Penerjemah/Penafsir AlQuran, *Al Quran dan Terjemahannya*
Departemen Agama, Jakarta, 1971.

Yulikuspartono, *Pengantar Logika dan Algoritma*, Andi, Yogyakarta, 2004.

Ansar, "<http://www.e-learning.daeng.web.id/uin/multimedia>", (26 November 2011).

Bambang Sigit, *Pengembangan Pembelajaran dengan Menggunakan Multimedia Interaktif untuk Pembelajaran yang Berkualitas*,
<http://luarsekolah.blogspot.com.pdf> (26 November 2011)



LAMPIRAN

Source Code

1. Login M-learning

```
package com.mobile.learning;

import mobile.config.*;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.io.OutputStreamWriter;
import java.net.HttpURLConnection;
import java.net.URL;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Intent;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class login extends Activity {
    public koneksi linkurl;
    String SERVER_URL;
    private Button login;
    private EditText username, password;
    public ProgressDialog progressDialog;

    //private TextView notif;
    /** Called when the activity is first created. */
    @Override
```

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.signin);

    login = (Button) findViewById(R.id.login);
    username = (EditText) findViewById(R.id.uname);
    password = (EditText) findViewById(R.id.passwd);

    //final TextView notif = (TextView) findViewById(R.id.tv_error);

    login.setOnClickListener(new OnClickListener() {

        public void onClick(View v) {

            // TODO Auto-generated method stub

            String Re;
            String mUsername = username.getText().toString();
            String mPassword = password.getText().toString();

            Re=tryLogin(mUsername, mPassword);

            Log.d("Check", "Here");
            Log.d("Re", Re);
            String temp_check=Re.trim();
            if(temp_check.equals("1"))
            {
                String nama = username.getText().toString();
                Intent newIntent = new Intent(login.this, halamanUtama.class);
                String txtnama = String.valueOf(nama);

                //membuat Bundle
                Bundle bundle = new Bundle();

                //menentukan parameter Bundle (id,isi) --> id=nama dan isinya
                //adalah variabel dari txtnama
                bundle.putString("nama", txtnama);

```

```

        //menambahkan bundle pada intent
        newIntent.putExtras(bundle);
        startActivityForResult(newIntent, 0);
        //notif.setText("SUKSES");
    }
    else
    {
        showDialog("Maaf", "Username Atau Password Salah !");
    }
    }
    });
}

protected String tryLogin(String mUsername, String mPassword)
{
    Log.d(" TryLoginCheck ", "Here");
    HttpURLConnection connection;
    OutputStreamWriter request = null;

    URL url = null;
    String response = null;
    String temp=null;
    String parameters =
"username="+mUsername+"&password="+mPassword;

    System.out.println("User Name"+mUsername+"\n"+"password"+mPassword);
    Log.d("Parameters",parameters);
    try
    {
        ;
        linkurl = new koneksi("/login.php");
        SERVER_URL = linkurl.getUrl();
        //"http://182.6.232.37:80/ta/login.php"
        url = new URL(SERVER_URL);
        connection = (HttpURLConnection) url.openConnection();
        connection.setDoOutput(true);

```

```

        connection.setRequestProperty("Content-Type", "application/x-www-form-urlencoded");
        connection.setRequestMethod("POST");

        request = new OutputStreamWriter(connection.getOutputStream());
        request.write(parameters);
        request.flush();
        request.close();
        String line = "";
        InputStreamReader isr = new
        InputStreamReader(connection.getInputStream());
        BufferedReader reader = new BufferedReader(isr);
        StringBuilder sb = new StringBuilder();
        while ((line = reader.readLine()) != null)
        {

            sb.append(line + "\n");
        }
        temp=sb.toString();
        Log.d("Temp",temp);
        // Response from server after login process will be stored in
        response variable.
        response = sb.toString();
        Log.d("Response",response);
        Log.d("Sb Value",sb.toString());
        isr.close();
        reader.close();

    }
    catch(IOException e)
    {
        Toast.makeText(this,e.toString(),0).show();
    }
    // Log.d("Response",response);
    return response;
}

```

```

private void createDialog(String title, String text) {
    AlertDialog ad = new AlertDialog.Builder(this)
        .setPositiveButton("Ok", null)
        .setTitle(title)
        .setMessage(text)
        .create();
    ad.show();
}
}

```

2. Halaman Utama M-learning

```

package com.mobile.learning;

import mobile.course.*;
import mobile.berita.*;
import mobile.download.*;
import mobile.tugas.*;
import mobile.nilai.*;
import mobile.about.*;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.TextView;

public class halamanUtama extends Activity {
    public void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.welcome);

        Button ibCourse = (Button) findViewById(R.id.ibKonten);
        Button ibDonlot = (Button) findViewById(R.id.ibDonlot);
    }
}

```

```

Button ibTugas = (Button) findViewById(R.id.ibTugas);
Button ibNilai = (Button) findViewById(R.id.ibNilai);
Button ibBerita = (Button) findViewById(R.id.ibBerita);
Button ibAbout = (Button) findViewById(R.id.ibAbout);
Bundle bundle = this getIntent().getExtras();

//Mengambil parameter 1 yaitu kode = nama
final String param1 = bundle.getString("nama");
TextView tv = (TextView) findViewById(R.id.welText);
TextView tv_logOut = (TextView) findViewById(R.id.logOut);

tv.setText("Selamat Datang -" +param1+ "- di M-Learning TIK");

tv_logOut.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent logout = new Intent(halamanUtama.this,
login.class);
        startActivity(logout);
    }
});
ibCourse.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        String namauser = param1;
        Intent newIntent = new Intent(halamanUtama.this, course.class);
        String txtnama = String.valueOf(namauser);
        //membuat Bundle
        Bundle bundle = new Bundle();

        //menentukan parameter Bundle (id,isi) --> id=nama dan isinya adalah
variabel dari txtnama
        bundle.putString("namauser", txtnama);

```

```
//menambahkan bundle pada intent
newIntent.putExtras(bundle);
startActivityForResult(newIntent, 0);
    }
});
```

```
ibDonlot.setOnClickListener(new OnClickListener() {
```

```
    public void onClick(View v) {
        // TODO Auto-generated method stub
        String namauser = param1;
        Intent newIntent = new Intent(halamanUtama.this, download.class);
        String txtnama = String.valueOf(namauser);

        //membuat Bundle
        Bundle bundle = new Bundle();

        //menentukan parameter Bundle (id,isi) --> id=nama dan isinya adalah
        variabel dari txtnama
        bundle.putString("namauser", txtnama);

        //menambahkan bundle pada intent
        newIntent.putExtras(bundle);
        startActivityForResult(newIntent, 0);
    }
});
```

```
ibTugas.setOnClickListener(new OnClickListener() {
```

```
    public void onClick(View v) {
        // TODO Auto-generated method stub
        String namauser = param1;
        Intent newIntent = new Intent(halamanUtama.this, pilihTugas.class);
        String txtnama = String.valueOf(namauser);

        //membuat Bundle
```



```

Bundle bundle = new Bundle();

//menentukan parameter Bundle (id,isi) --> id=nama dan isinya adalah
variabel dari txtnama
bundle.putString("namauser", txtnama);

//menambahkan bundle pada intent
newIntent.putExtras(bundle);
startActivityForResult(newIntent, 0);
    }
});

```

```

ibNilai.setOnClickListener(new OnClickListener() {

```

```

    public void onClick(View v) {
        // TODO Auto-generated method stub
        String namauser = param1;
        Intent newIntent = new Intent(halamanUtama.this, nilai.class);
        String txtnama = String.valueOf(namauser);

        //membuat Bundle
        Bundle bundle = new Bundle();

        //menentukan parameter Bundle (id,isi) --> id=nama dan isinya adalah
        variabel dari txtnama
        bundle.putString("namauser", txtnama);

        //menambahkan bundle pada intent
        newIntent.putExtras(bundle);
        startActivityForResult(newIntent, 0);
    }
});

```

```

ibBerita.setOnClickListener(new OnClickListener() {

```

```

    public void onClick(View v) {
        // TODO Auto-generated method stub

```

```
        Intent intentBerita = new Intent(halamanUtama.this,
berita.class);
        startActivity(intentBerita);
    }
});

ibAbout.setOnClickListener(new OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent intentAbout = new Intent(halamanUtama.this,
about.class);
        startActivity(intentAbout);
    }
});
}
```



RIWAYAT HIDUP PENULIS



Sutrisno akrab dengan panggilan Inho, lahir di Pandang-pandang, 26 Maret 1988. Anak ketiga dari tiga bersaudara. Anak dari pasangan Bapak Muhtar M, S.Pd. dan Ibu St. Norma A, SE. ini lebih senang bergaul dengan para rakyat jelata. Mempunyai prinsip hidup yaitu “akhir dari kesabaran dan perjuangan adalah kematian”.

Pendidikan yang pernah ditempuh :

1. Sekolah Dasar pada tahun 1994-2000 di SD Impres Sungguminasa.
2. Sekolah Menengah Pertama pada tahun 2000-2003 di SMP Negeri 2 Sungguminasa.
3. Sekolah Menengah Kejuruan pada tahun 2003-2006 di SMK Negeri 2 Makassar.
4. Melanjutkan pendidikan pada Universitas Islam Negeri (UIN) Alauddin Makassar, fakultas Sains dan Teknologi pada tahun 2007, mengambil jurusan Teknik Informatika.

Organisasi yang pernah diikuti :

1. HMJ Teknik Informatika pada tahun 2008-2009.
2. BEM Fakultas Sains dan Teknologi pada tahun 2009-2010.
3. Ketua LPP Fakultas Sains dan Teknologi pada tahun 2011.